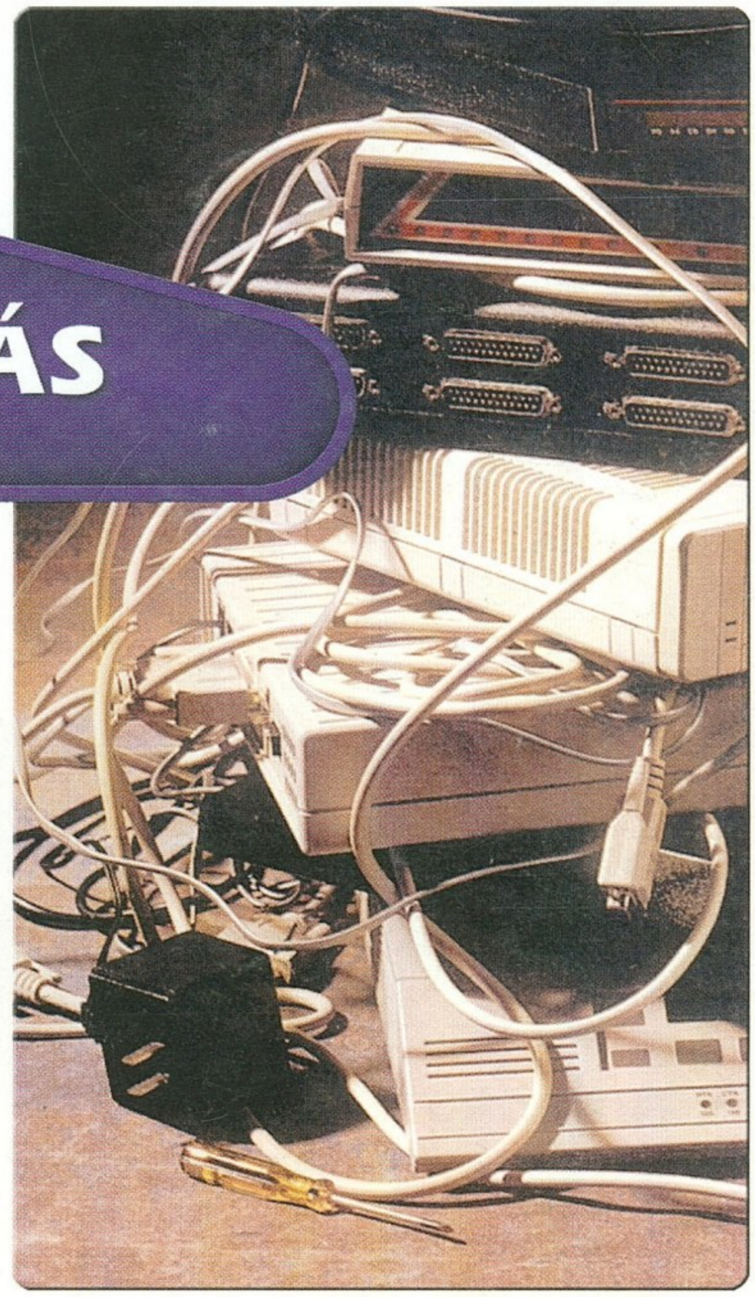
Magyar számítástechnikai folyóirat CD-melléklettel Ára 699 Ft

XVIII. évfolyam 6. szám 2000. június

ÚJ ALAPLAP

A hónap témája:

KOMPATIBILITÁS





Szelídítsünk pingvint - I.

Programozástechnika

Változatok a szerelemre

Vírusőrjárat

Üzlet az üzlet a neten

Kuk@cológia

A Microsoft-per margójára

Nyílt tér

Rendszer és a rendszeretet

Közkincs

Magyarul diktálni a gépnek

Műhely

00006
9771217759005

ÚJ ALAPLAP

A Mikroszámítógép Magazin és az Alaplap hagyományait folytató magyar számítástechnikai folyóirat

Megjelenik havonta, CD-melléklettel

Főszerkesztő:
Faklen Pál

Szerkesztő:
Jakab Ágnes

A szerkesztőbizottság tagjai:
Aszalós László, Bánó György, Feleki Zoltán, Galántai Zoltán, Herczeg József, Kádár Zsolt, Kovács Attila, Mákos András, Nagy Tamás, Pogány Csaba, Sándor Gábor, Simay Endre István, Szappanos Gábor, Szondi Egon János, Vargha Dénes, Vékony Tamás

Szerkesztőség és kiadó:
1539 Budapest, Pf. 571
VI., Dózsa György út 84/b
Telefon: 322-4417, 322-5238
Fax: 351-8015
E-mail: alaplap@mail.datanet.hu
Weblap: http://www.alaplap.hu

Felelős kiadó:
Faklen Pál

Terjesztés:
Megyes Zsuzsanna

Hirdetésszervezés:
Árvai Katalin, Galyasi Hedvig, Pap Katalin

Külföldi hirdetések:
PubliciTeam
Reklám- és Médiaügynökség
1537 Budapest I., Márvány u. 17.
Telefon: 356-1182 Fax: 375-3539

A kiadó a hirdetések tartalmáért és a nyomdakészen kapott hirdetések formájáért (és helyesírásáért) nem vállal felelősséget

Példányszámadatok hitelesítése:
Magyar Terjesztésellenőrző Szövetség
MATESZ

Ez a szám 9000 példányban jelent meg

Nyomtatás:
Zalai Nyomda Rt, Zalaegerszeg
Felelős vezető:
Czirkl György vezérigazgató

Terjeszti:
a Lapker Rt, a Hírker Rt, az NH Rt, az MP Rt LHI és számos számítástechnikai szaküzlet

Előfizethető a kiadónál:
Új Alaplap Kiadói Kft,
1539 Budapest, Pf. 571
Bankszámlaszám:
OTP 11706016-20788599

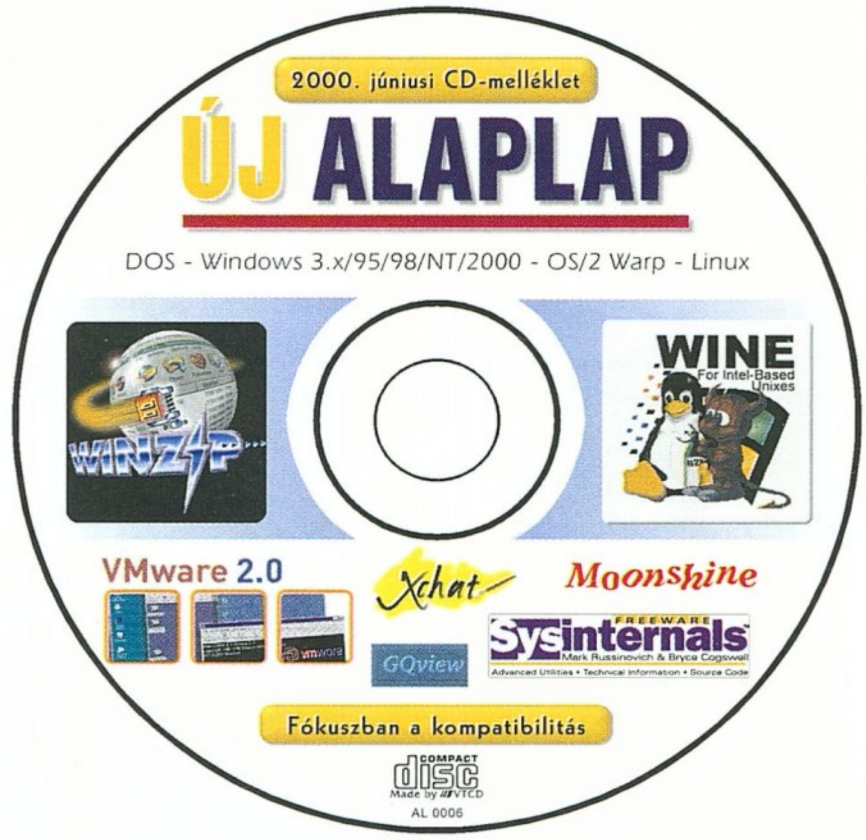
A lap példányonkénti ára: 699 Ft
Évi előfizetési díj: 6996 Ft

Külföldi előfizetés díja:
6996 Ft + postázási költség

HU ISSN 1217-7598

TARTALOM

XVIII. ÉVFOLYAM 6. SZÁM, 2000. JÚNIUS

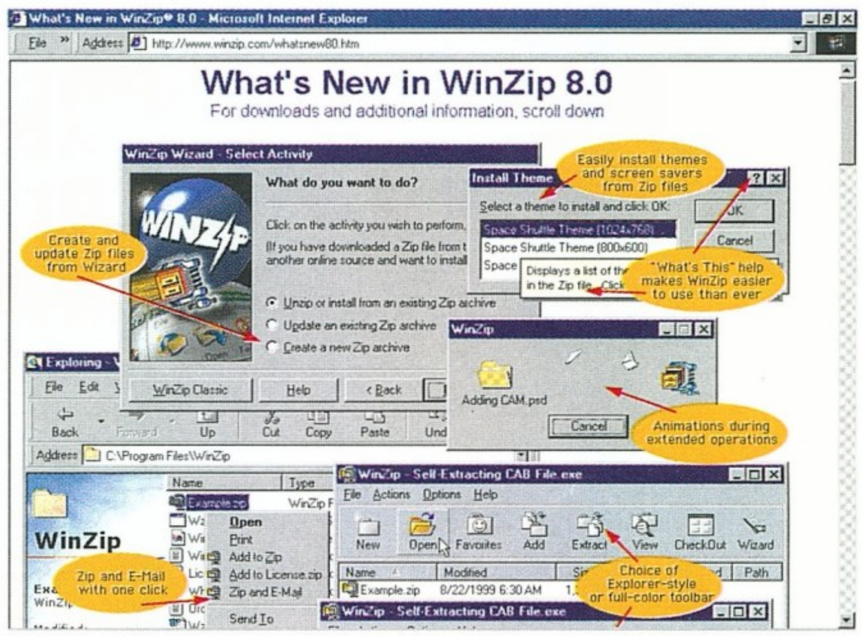


Fókuszban a kompatibilitás

FOKUSZ	
EMU	Emulátorok különböző platformokra
C64	Commodore 64 emulátorok PC-re
DOSEMU	Dosemu 1.0.0, DOS-emulátor Linuxra
MESS	Multi-Emulator Super System 0.36
VMWARE	PC-emulátor különböző platformokra
WILLOWS	Willows Twin Libraries a Windows API-k megvalósításához különböző platformokon
WINE	Wine, Win3.x és Win32 API emulátor
ZX	XZX v2.9.2, X11 alapú ZX Spectrum emulátor Unixra
RBROWN	Ralf Brown megszakításlistája
W2KHCL	Windows 2000 Hardware Compatibility List, a W2k által támogatott hardverek listája
WINDRIV	WinDriver, multiplatformos driverek fejlesztését lehetővé tevő szoftver különböző platformokra

Lapraforgó

LAPFORG	
BEOS	Ikonok a Be operációs rendszerhez (26. oldal)
JAVA	Illusztrációk a Java tanfolyamhoz (64. oldal)
KOZKINCS	Rendszer és rendszeretet (40. oldal)
AIDA	Aida v1.00-0511, hardver- és szoftvertesztelő program
CONTIG	Contig v1.3, fájlok töredezettségét megszüntető program
ERDCOMM	ERD Commander, NT betöltése és javítása floppyról
FILEMON	Filemon v4.28, fájlfigyelő (monitorozó) program
HANDLEEX	HandleEx v2.23, segédprogram a futó alkalmazások és az általuk lefoglalt erőforrások megtekintésére
PORTMON	Portmon v3.01, a soros és párhuzamos portokon keresztüli forgalom figyelésére alkalmas program
REGMON	Regmon v4.24, a registryt folyamatosan figyelő program
OS2	Nemcsak OS/2 ... (32. oldal)
FIXPAK	FixPak 13 az OS/2-höz, CD-ről is telepíthető változat
STELLAR	Stellar Frontier, stratégiai játék OS/2-re és Windowsra
URL	A lapban hivatkozott URL címek listája



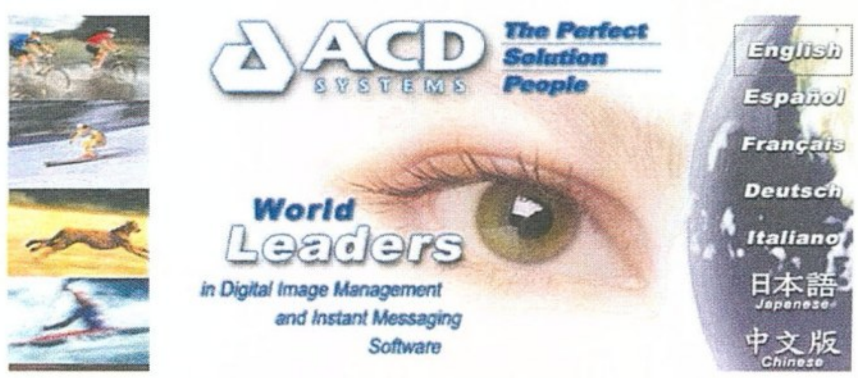
Szerszámosláda

SZERSZAM	
DOS	DOS-os alkalmazások
WIN3X	16 bites windowsos alkalmazások
WIN9X	32 bites windowsos alkalmazások
LINUX	Linuxos alkalmazások
OS2	OS/2-es alkalmazások
FESZER	Gyakran szükséges programok
VIRUS	Vírusirtók



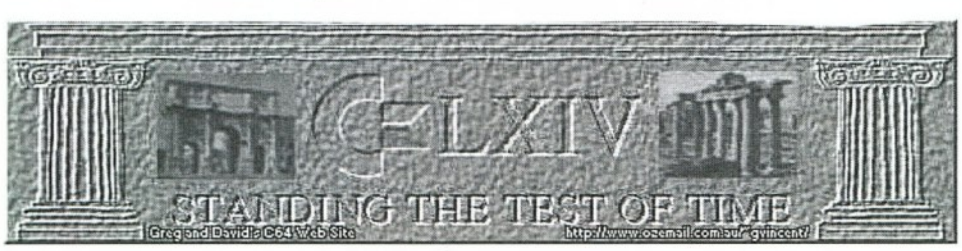
Vendégoldal

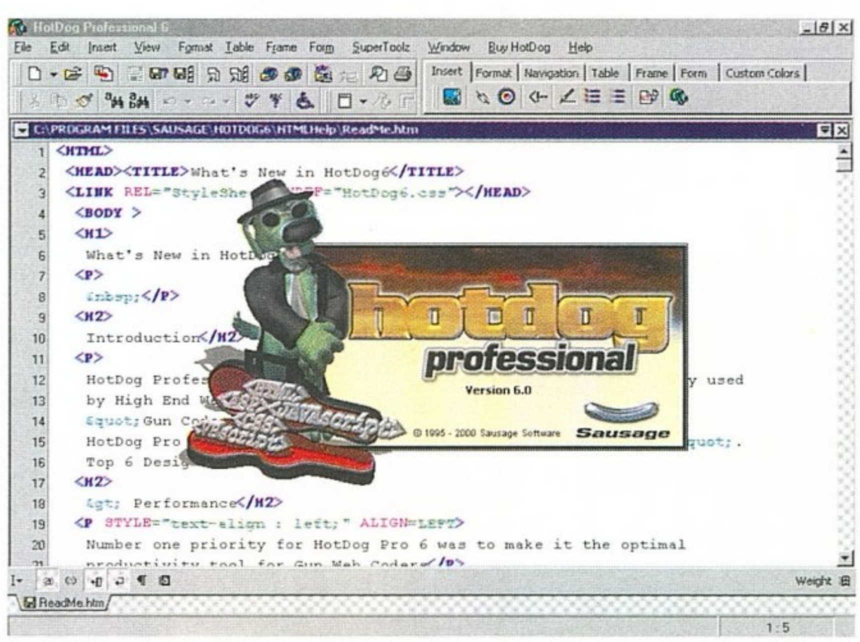
VENDEG	
ACDSEE	ACDSee képnézegető, plug-inokkal
DELCOMP	Delphi komponensek
DOKISOFT	EasyBackup 4.1, archiváló program (Dokisoft)
IBM	NetRexx 1.160
KDE	Kommon Desktop Environment 2.0 (béta 1, kód 1.90)
MICROSFT	Hibajavítások, dokumentációk a Microsofttól
FP2SDK	Programozási segédlet az új FrontPage-hez
IEPATCH	Hibajavítás az Internet Explorer 4-eshez, 5-öshöz
O2KPATCH	Az Office 2000 UA Control biztonsági hibát javító csomag
O2KSP	Office 2000 SR-1 javítócsomag
OUTLOOK	Javítás az Outlookhoz a vírusok terjedése ellen
PERFTUNE	Windows 2000 Performance Tuning v1.0
NETREXX	Kádár Zsolt NetRexx tanfolyama (1-7. rész)
NETSCAPE	Netscape Communicator 4.73 Linuxra és Windowsra
NRGCOM	Horizont, számlakészítő modul, (NRG-COM Kft)
SUN	Java 2 SDK, Standard Edition, v1.3 Windowsra Java 2 Runtime Environment Windowsra Java 2 SDK, v1.3 dokumentáció Forte for Java, Community Edition 1.0, fejlesztőkörnyezet (IDE) Linuxra és Windowsra
SYMANTEC	
VBUSTER	VirusBuster, vírusirtó program
WXPYTHON	Grafikus (GUI) eszközkészlet a Python nyelvhez



Játékvár

JATEK	
BREAKOUT	Falbontó tenisz
MEMORY	Memory 1.0, memóriajáték (Csordás Árpád)
TABLA	Táblajátékosok fóruma





Még önmagával sem...

Ambivalens háztáji kompatibilitás

A kompatibilitás kérdései aligha kerülhetők meg egy olyan számítógépes világban, ahol megannyi hardver és még több szoftver igen sok variációja egyidejűleg létezik, ezért különbözőképpen összerakott gépeken és változatos platformokon van lehetőségünk (vagy vagyunk kénytelenek) dolgozni. A régebbi gépek nem válnak egyik napról a másikra használhatatlanná, csak azért, mert újabbak jelentek meg a piacon, és a szoftvereknek sem kell automatikus pénztemetőkké válniuk. Ki gondolta volna, hogy a 32 bites korszakban a már muzeálisnak tekinthető 8 bites gépek programjai továbbra is közkézen forognak majd, sőt folyamatosan továbbfejlesztik őket — ahogy azt lapunk közelmúltban közölt sorozata jól bemutatta. (Tarján Richárd írásai a Spectrumról, Új Alaplap, 1999/12., 2000/2. és 3. szám.)

A kompatibilitásra való törekvés átszövi az egész számítástechnikát. De ugyanúgy a kompatibilitástól való elszakadás is. A szakma nagyon ambivalens viszonyban van ezzel. Pontosabban nem is maga a szakma, hanem inkább a szakma profitérdekeltsége. Egyrészt ugyanis mindenkire érvényes alapfeltétel a bizonyos szintű szabványosodás, mert enélkül az eszközök terjedése és használata leküzdhetetlen akadályokba ütközne.

Másrészt viszont a gyártók azáltal (is) igyekeznek a többiek elé kerülni, hogy (az okvetlenül szükséges azonos vonásokat persze megtartva) saját szabványokkal igyekeznek a versenytársak termékei számára nem kompatibilis környezetet, „védett piacot” teremteni. (A szabvány kifejezést most nem jogi, hanem a szélesebb köznapi értelemben, esetenként a formátum szinonimájaként is használva.)

Amint elterjedtek a személyes használatra szánt számítógépek, egyszerűbbé kellett tenni a programok futtatását a különböző gépeken.

A programozás terén erre elsősorban a Basic kínált lehetőséget. Az eredetileg oktatási célra kifejlesztett nyelv egyszerűsége folytán a kódolásban kevésbé járatos felhasználók is alkalmas eszközre leltek benne.

A hordozhatóság ebben a formában még elsősorban a kinyomtatott szöveg újbóli begépelését jelentette. Mivel a gépcsaládok (Commodore, Enterprise, Sinclair stb.) más-más Basic „dialektust” beszéltek, begépeléskor korrigálni kellett a kisebb különbségeket, az alapalgoritmusok viszont általában változatlanul átvihetők voltak.

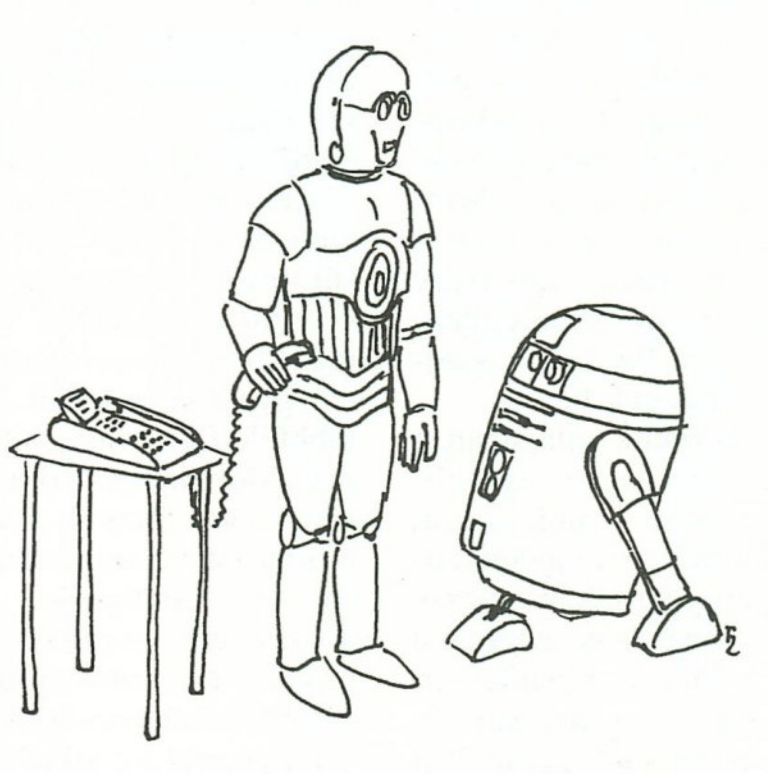
Igazán hatékonyan persze ezeket az eszközöket is gépi kódban lehetett programozni, amit az igényesebb alkalmazások, jobb játékok készítői ki is használtak, de ez egyúttal már gátolta a programok átvitelét egyik családból a másikba, sokszor még azonos családon belül is egy másik típusra. A gépek elterjedésével azonban megjelentek olyan illesztőprogramok, amelyek lehetővé tették a magnókazettára mentett állományok értelmes beolvasását más géptípuson is.

Hasonló lehetőségeket keresünk, amikor a régi gépekre írt programokat szeretnénk napjaink elterjedt PC-in feléleszteni: segédprogramok, emulátorok teszik lehetővé a 8 bites gépek programjainak futtatását. Egy részük akár több régi géptípus környezetét is tudja utánozni. Az emulátorok általában szabad forráskódúak, és több platformra átírhatók, ezért a DOS/Windows mellett Linux és OS/2 operációs rendszeren, illetve Macintosh gépeken is hozzáférhetők.

Kortársak emulálva

Nem nélkülözhetjük az emulátorokat akkor sem, ha napjaink programjait szeretnénk futtatni más, „nem kompatibilis” gépeken. Legtöbbször ugyanis bináris formában meglévő szoftverek futtatását kell megoldani. Már az is probléma lehet, ha például Macintosh és Intel alapú rendszerek között kell adatokat átvinni, bár ilyesmire elég korán megírták a felírási formátumok közötti különbségek áthidalására szolgáló (például floppybeolvasó) segédprogramokat. A programok, a bináris állományok futtatásához ennél természetesen sokkal több kell, de erre is van megoldás.

Elkészültek és ma is folyamatosan fejlődnek azok a programok, amelyek lehetőséget teremtenek a programok más platformokon történő futtatására. A Macintosh környezet számára készült a RealPC szoftver, amelynek elsődleges célja a kedvelt DOS programok „im-



— R2D2! Egy telefax van a vonalban. Beszélnél vele?



portálása" volt. (Lapunkban 1998 októberében írtunk róla.) Az Insignia Solutions készítette, hamar megcélozva a teljes WIntel környezet elérését Macintosh platformról, de eladta a RealPC–SoftWindows csomagot az FWB Software-nek (http://www.fwb.com).

Folyamatosan bővül a DOS és Windows programok futtatásának lehetősége napjainkban Linuxon is. Az erre szolgáló programok forráskód szintén rendelkezésre állnak, és lehetőséget adnak a széles körű emulációra, vagyis az alaprendszer mellett a gépünkre telepített egyéb eszközöket is igénybe vehetjük. Ez különösen fontos a windowsos programoknál, mert így lehetőség van a DLL-ek használatára. Az OS/2-re írt emulátorok sorsát pedig lapunkban kísérhetik figyelemmel az Alternatíva rovat „Nem csak OS/2 ..." sorozatában.

A más platformokra való átlépés igénye a PC-s világ mellett az összetett hálózatokban is megvan. Több hálózatos operációs rendszerben megoldották például, hogy használni lehessen a Windows NT grafikus környezetét. Nem is mindig emulációval, hanem úgy, hogy az egyébként más processzorarchitektúrára épülő eszközbe Intel processzoros bővítőkártyát építettek be.

Az összetett hálózatok világában a szerveren futó alkalmazáshoz egy kliensgépről gyakran úgy férünk hozzá, hogy a munkaállomáson tulajdonképpen csak a képernyőtartalom jelenik meg, és a vezérlőutasítások utaznak a hálózaton (Citrix, MS TerminalServer, SCO Tarantella stb.). Itt a kompatibilitást többnyire nem az adott gépre átírt kóddal oldják meg, hanem a rendszerből csak a kliensprogramot ültetik át különböző platformokra, kikerülve ezáltal az operációs rendszerek belsejének aknazárját és a verzióütközéseket. Ha pedig azt Jolly Joker nyelven, például Java kódban írták meg, akkor a feladat áthárítható a Java virtuális gépre (Java Virtual Machine, JVM).

Verziók egymás közt

Kompatibilitási problémákért persze nem kell „a szomszédba" menni. Elegendő ehhez az operációs rendszer frissítése, verzióváltása. Az XT-n indult, majd a 32 bites processzorok világába átörökített MS-DOS verziók rendszerállományai például nem szerették a különböző verziójú komponensek keveredését. Erről bárki meggyőződhet, ha a rendszerindító floppyn megpróbálja összeházasítani az eltérő verziókból származó IO.SYS, MSDOS.SYS, COMMAND.COM állományokat. De ez valószínűleg így van rendjén a rendszerindításhoz szükséges állományok esetében.

Kevésbé érthető, hogy a memóriamenedzser programok (EMM386.EXE stb.) miért érzékenyek annyira a DOS verzióira, miközben hasonló szerepű más komponensek (amilyen a játékoknál gyakran használt DOS4GW.EXE) többféle DOS változattal is jól működnek. Memóriakezelőkre pedig szükség van, mert nélkülük a kiterjesztett memóriát intenzíven használó alkalmazások nem futtathatók.

Az operációs rendszerek gyártói által produkált verziófüggőség sújtotta leginkább a felhasználókat, de éltek ezzel a fegyverrel az alkalmazások készítői is, beépítve programjaikba a DOS belső verziójának ellenőrzését. Tették ezt néha okkal, de többnyire ok nélkül. Ki lehetett ugyan cselezni a SETVER.EXE program használatával, mely a verzióolvasásra vonatkozó megszakításkérést elfogva a megfelelő értéket adta vissza, feltéve persze, hogy az igény alacsonyabb verziószámú DOS-ra vonatkozott. A DOS verziószámának lekérdezését pedig egy belső függvény teszi lehetővé.

A DOS függvényeinek elérése az alkalmazások készítésekor más esetekben is hasznos lehet. Ehhez Ralph Brown jóvoltából rendelkezésünkre áll egy megszakításlista, melynek alapján kevésbé dokumentált funkciókat is használhatunk. A változó verziók világában talán éppen ezek a funkciók őrizték meg legjobban a visszamenőleges kompatibilitást.

A Windows konfliktusai

Nem változtatott a helyzeten a Windows megjelenése sem, mert annak első elterjedt verziócsaládja, a Windows 3.x tulajdonképpen grafikus héjként viselkedett. Futtathattunk DOS-os alkalmazásokat, és a DOS-ból indíthattuk el magát a grafikus héjprogramot is. Közben a 16 bites Windows alatt futtatott programok számára ugyanúgy elérhető maradt számos DOS funkció. Hasonló lehetőség megmaradt eleinte a Windows 95-ben is: a betöltődés közben megnyomott F4 vagy az utána megnyomott F8 billentyű által előhívott menüből startolhat az előző operációs rendszer, a DOS, amennyiben a Win95-öt nem teljesen szűz gépre telepítetették.

A WIntel platformokon jelentős váltás volt a kettős betöltődés (dual boot) támogatásának megszüntetése a Windows 95 OSR2 változata után, bár ezt megelőzően is folyton voltak verziókövetési problémái a Windows rendszer alatt működő DOS-nak. Tapasztalataim szerint a Win9x programok alapesetben úgy módosítják az indítófájlokat, hogy azokba bekerül a már említett SETVER.EXE használata. Ez persze annak a következménye is lehet, hogy az alábbiakban kifejtett okból mindig egy működő MS-DOS 6.22-re telepítettem a Windowst.

A DOS-os programok egy részét a Win9x nem hajlandó futtani, a Windows menüjéből választott parancssoros indítás esetén sem. Emögött részben a Windows rezidens alapprogramjainak nagyobb tárigénye húzódik meg. Ha viszont DOS 6.22-es floppyról indítjuk a gépet, akkor a C:\Win9x könyvtárban található programok verzióütközést jeleznek. Szükség van tehát egy teljesen DOS-os, tetszőlegesen paramé-

terezhető futtatókörnyezet kialakítására. Ennek egyik részét a DOS rendszerlemez (bootfloppy) adja, azon vannak a szükséges CONFIG.SYS és AUTOEXEC.BAT fájlok, hogy a DOS-t különböző memóriakonfigurációk mellett elindíthassuk. Emellett célszerű helyet szorítani a segédprogramoknak a menet közben esetleg fellépő problémák orvosolására (fdisk, format, mem, sys stb.). A rendszer másik részét igénybe vehetjük a merevlemezről, ha arra előzetesen telepítettük a DOS-t. Szükség lehet még olyan DOS komponensekre is, mint a memóriakezelők (HIMEM .SYS, EMM386) és a perifériakezelők (hang, egér, külső merevlemez, nyomtató stb.).

„Random" rendszer

A Windows futtatásakor tapasztalható kompatibilitási problémák másik köre a látszólag azonos programrendszerek belső eltéréséből fakad. A gondok fő forrásai a DLL komponensek. Egyes programok telepítőkészlete ugyanis a meglévő, korábban telepített DLL állományt mindenféle (kárelkerülő) ellenőrzés nélkül lecseréli másik (neki tetsző) változatra.

A „csere" révén kapott DLL sokszor inkompatibilis a korábban telepített programokkal. Utólag elég nehéz ezeket a hibákat korrigálni, mert a fájlok neve nem tartalmaz verzióinformációt (16 bites környezetben a 8+3-as korlát ezt nem is mindig tenné lehetővé). A futtatható fájlok „VERSIONINFO" erőforrásában ott vannak a szükséges adatok, de ha megtekintésükre nincs alkalmas eszközünk, akkor csak a fájl mérete és keletkezési időpontja ad bizonyos támpontot.

A Windows 3.x korszak verziókáoszának generálásában élen jártak magának a Windows rendszernek a készítői. A piactarolási megfontolásokból kiérleletlenül piacra dobott Windows 3.1-nek a hibáit csak suttyomban javítgatták ki, és a forgalomba került telepítőkészletekbe mindig belerakták a már kijavított elemeket. Aki megfelelő háttérmentés nélkül vágott bele valamilyen frissítésbe, az végül szinte véletlenszerűen jól vagy rosszul működő rendszert talált a gépén. A különböző nyelvi verziók egyébként névben és olykor méretben is azonos állományainak összeférhetetlensége pedig betetőzte a káoszt.

A 32 bites rendszerekben valamelyes javulást eredményezett a Registry. Abba (elvben) már elegendő egyszer beírni a komponensre vonatkozó eligazítást, és onnan kezdve a rendszernek mindig pontosan tudnia kell(lene), hogy mihez mit válasszon, és mit hol keressen. Vannak azonban helyzetek, amikor ez nem segít, mert egyes telepítőprogramok a Registry frissítése nélkül írnak felül fájlokat, lévén, hogy a Windows 9x máig őriz 16 bites alkotóelemeket, amelyek kevésbé védettek. Ilyen szempontból továbbra is esettanulmány lehetne az angol nyelvű Windowsra telepített 16 bites magyar nyelvű Office csomag. Pedig erre igencsak szükségünk van, amikor ékezetesen szeretnénk látni betűinket.

Ehhez kapcsolódik egy sajátos kompatibilitási probléma, melynek alapkérdéseivel, a honosítással foglalkozott 2000. áprilisi számunkban a hónap témája összeállítás. Aki próbált már ékezethelyesen írt szöveget átvinni a különböző MS Word verziók között, az tapasztalhatta, hogy arra az RTF formátum a legalkalmasabb. Ékezetes betűink ott ugyanis akkurátusan be vannak kódolva a megfelelő hexadecimális értékkel. Igen ám, de sajátos (és sajnálatos) módon a 32 bites Word ezeket a betűket a dokumentumban hol unikódos, hol normál (256 alatti ASCII) karakterként tárolja, és ezt a vágólapra helyezve is érvényesíti. Ha tehát sima szövegállományra van szükségünk, hiába élünk a Word 97 saját konvertálási lehetőségével, vagy hiába próbáljuk a vágólapon keresztül kimásolni a szöveget, az eredmény az „ő" és „ű" betűk ékezeteinek elvesztése, o és u betűkre történő véletlenszerű leborotválása lesz. Ilyenkor a legegyszerűbb megoldás keresni egy 16 bites rendszert, és annak Wordjében elvégezni a konverziót.

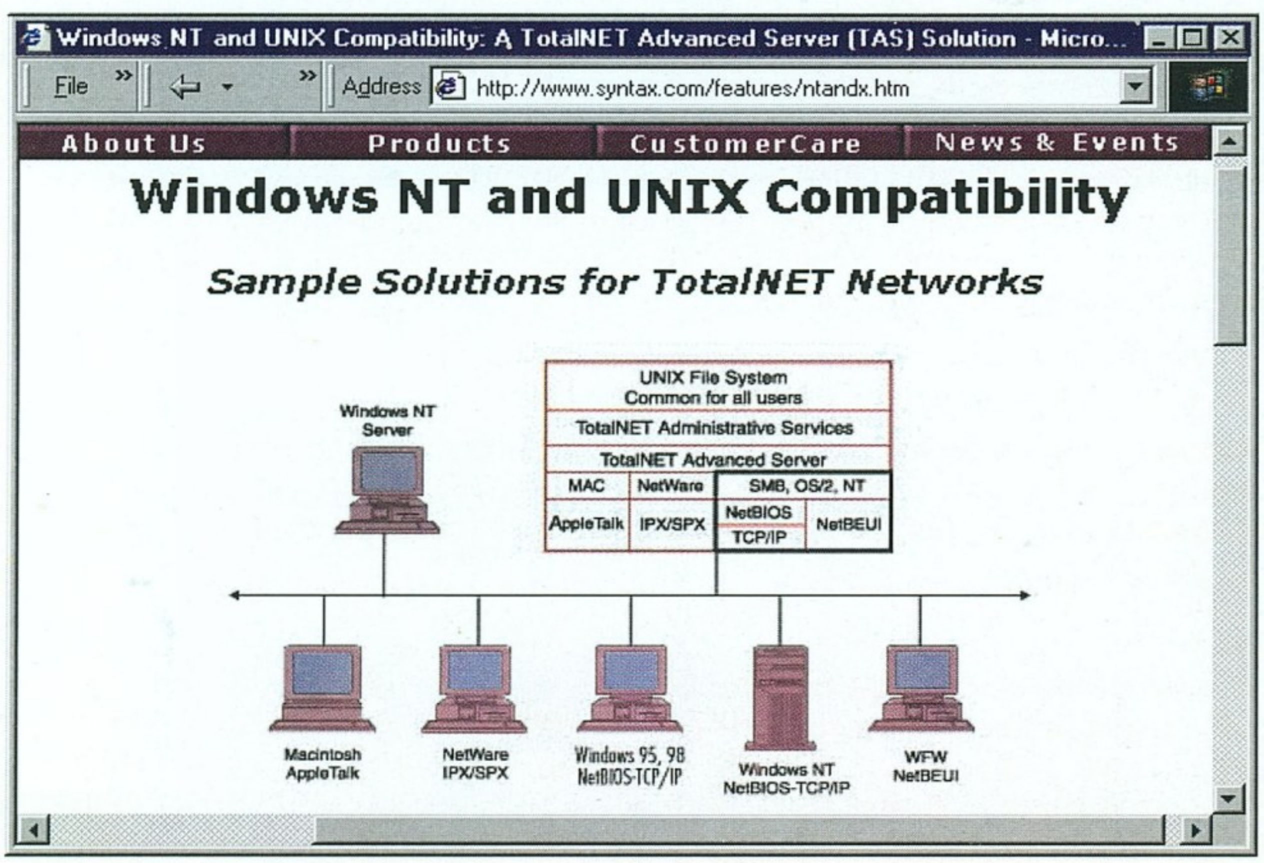
Nyitás parancsra?

A verziók belső ütközései miatt keletkező „DLL-problémák" kiküszöbölésére a Microsoft bevezetett néhány újítást a Windows 2000-ben. Ezek hatékonyságát azonban — a korábbi verziókkal szerzett tapasztalatok alapján — csak az idő fogja megmutatni, amikor majd az alkalmazások és programfrissítések a gyakorlatban vizsgáznak.

A verzióütközések, könyvtáreltérések stb. kiköszöbölésére van persze olyan lehetőség is, amelyről a Microsoft nem akar tudomást venni. Ha a teljes rendszer, vagy annak legalább a kritikus része forráskódban vagy binárisan előfordított állapotban hozzáférhető, akkor a 32 bites és a készülő 64 bites rendszereken nincs akadálya annak, hogy már a csomagok nevében is szerepeljenek verzióinformációk. Ezek birtokában ránézésre eldönthető az elsőség kérdése és a frissítés szükségessége. A rendszer számára pedig egy azonos kapcsolófájl biztosíthatja, hogy ne kelljen mindig mindent újrafordítani. Valahogy úgy, ahogy ezt a Linux intézi a LIB-ek esetében. Pedig ez a nem is Linux-specifikus, hiszen a Windows munkaasztalán található ikonok jelentős része mögött ugyanilyen kapcsolófájlokat (shortcut) találunk.

A Microsoft ellen megszületett ítélet konkrét szankcióinak egyike a Windows forráskódjainak legalább részleges felszabadítása lehet. Az eredmény nem csupán a következetes verziókezelés lenne, hanem megvalósulhatna a kódok optimalizálása és a komponensek közti felesleges redundanciák kiszűrése is. És az a Windows már nem ez a Windows lenne.

Simay Endre István
endre_s@excite.com



INFORMÁCIÓKÉRÉS: 15 ▲

Ericsson-díjak tanárok számára

Az Ericsson Magyarország 1999-ben díjakat alapított az általános iskolában, szakközépiskolában, gimnáziumban és speciális tagozaton oktató matematika- és fizikatanárok munkájának elismerésére.

I. A 200 000 forinttal járó **„Ericsson a matematika és fizika tehetségeinek gondozásáért"** díjat minden évben 4–4 olyan matematika- és fizikatanár kaphatja meg, akinek tanítványai az 1996/97-es tanévtől kezdődően I., II. vagy III. díjat nyertek az alábbi tanulmányi versenyek valamelyikén: Középiskolai Matematikai és Fizikai Lapok versenyei; Varga Tamás, Arany Dániel matematikaversenyek; matematika vagy fizika OKTV-k; Öveges József, Mikola Sándor, Fényes Imre, Szilárd Leó fizikaversenyek; Nemzetközi Matematika vagy Fizika Diákolimpiák, Kürschák József matematikai tanulóversenyek; Eötvös Loránd fizikaversenyek.

II. A 150 000 Ft-tal járó **„Ericsson a matematika és fizika népszerűsítéséért"** díjat ugyancsak 4–4 matematika- és fizikatanár kaphatja meg évente. Az idei díj elnyerésének feltétele, hogy az 1999/2000-es tanévben tanítványai aktívan bekapcsolódtak a Középiskolai Matematikai és Fizikai Lapok pontversenyeibe (korábban az adott iskolából egyáltalán nem, vagy csak csekély eredménnyel versenyeztek a tanulók), illetve több éve elismerten sokat tesznek a tantárgyuk iránti érdeklődés felkeltéséért, a tantárgy megszerettetéséért.

A díjakat a Matfund Középiskolai Matematikai és Fizikai Alapítvány ítéli oda, a Bolyai János Matematikai Társulat és az Eötvös Loránd Fizikai Társulat Ericsson-díj bizottságainak ajánlása alapján.

A díjazottak személyére írásos javaslatot nyújthatnak be a szakmai és társadalmi szervezetek, a javasolt tanár tevékenységét ismerő kollégák vagy tanítványok. A javaslatokat a Bolyai János Matematikai Társulat, illetve az Eötvös Loránd Fizikai Társulathoz kell eljuttatni, legkésőbb 2000. október 1-jéig (1027 Budapest II., Fő u. 68.).

A díjbizottságok a beérkezett javaslatok és a versenyek eredményei alapján teszik meg ajánlásaikat a díjazottakra, 2000. november 1-jéig.

További információk:
Liptay Gabriella, kommunikációs manager
Ericsson Magyarország, Telefon: 437-7134
E-mail: gabriella.liptay@eth.ericsson.se

Szórólap szinten kompatibilis

„Vagy alusznak, vagy nem hallják...”

Compatibility: összeegyeztethetőség, összeférhetőség. Mint oly sok minden a számítástechnikában, természetesen ez a fogalom is sokkal tágabb értelmet nyert az évek során... Erről „számol el” ez az írás.

Amikor azt a szót hallom: kompatibilis, mindig régi „történetek” jutnak az eszembe.

A fekete doboz

A KGST idején a szocialista tábor számítógépei az ESZR sorozatból álltak. A Videoton is készített ilyet (R10, R11), meg különféle részegységeket. Már nem is emlékszem a monokróm, zöld terminál típusszámára. Gyakorlatilag azonban megfelelt egy IBM 327x terminálnak. Olyannyira, hogy éveken át használtunk jó néhányat az IBM 4341-4361-es pároshoz, sőt még a 9370-esekhez is.

A terminál egy csúf műanyag doboz volt, pici monitorral, külön billentyűzettel. A nagygéphez a 3270-es hálózathoz használt koaxon keresztül csatlakozott, néha egy közbeiktatott, szintén Videoton gyártmányú terminálvezérlővel. A belseje nem is hasonlított az eredeti IBM gyártmányokéra. A VT gépben több kártya dolgozott, 8080-as processzorok köré építve.

Az nem derült ki, hogy az IBM terminálokban pontosan mi is ketyeg, mert már akkoriban is voltak olyan trükkök, hogy több áramkört egybe tokoztak (afféle IC megoldás). De a kreatív magyar mérnököknek nem is kellett az eredeti belsejét megismerniük. Kimérték, hogy a kábelen milyen jelsorozatot vált ki valamilyen esemény (egyébként is elég jó dokumentációkhoz lehetett akkoriban hozzájutni, ha nem is mindig „egyenes úton”), és megalkottak egy olyan „fekete dobozt”, amely kifelé ugyanazt produkálta, mint a „példaképe”, sőt a bejövő jelek hatására is ugyanazt jelenítette meg a monitoron, amit egy valódi IBM.

Ez volt ám az igazi kompatibilitás, hisz az tulajdonképpen senkit nem érdekelt (a műszakiakon kívül), hogy mi van belül, és hogyan működik, ha egyszer minden külső eszközt (monitort, billentyűzetet) ugyanúgy lehet hozzá használni...

Kicsit később, amikor az IBM-kompatibilis PC kezdett terjedni, a Vidi beszállt ebbe a buliba is. Megszülettek a VT-s PC-k. Speciális, 113 gombos billentyűzettel, masszív, nehéz dobozzal, csúf kis mono monitorral. Akkoriban egy kiállításon nagyjából a következő beszélgetés zajlott a Videoton standjánál:

— Érdeklődnék a PC-jükről...

— Tessék!

— A szórólapon az áll: IBM PC-kompatibilis. De milyen szinten?

— ??? ... Szórólap szinten...

Azóta nálunk szállóige a „szórólap szinten”. Ha valami nem igazán akar beindulni ott, ahol a leírás és a „kompatibilitás” mián mennie kéne, akkor hamar kibukik valamelyikünkből: „Na, ez is csak szórólap szinten...”

Hardver

A PC-s világban a kompatibilitás valami ilyesmit jelent: ha egy program fut egy 8086/88-as alapú XT-n, annak a vele kompatibilis egyéb gépeken is futnia kell. Ez valamikor igaz is volt, hisz szinte minden PC ugyanazon a hardverkészleten alapult. Azután megindult a különféle klónok áradata. Ezek máig is azt hirdetik, hogy „kompatibilisak”, mégis folyton belefutunk olyan jelenségekbe, amelyek azt bizonyítják: nem teljesen. A videokártyáknál, hangkártyáknál gyakran ütközünk komoly akadályokba. Persze a problémát jól megírt driverkészlettel még el lehet űzni, de mégsem teljes a kompatibilitás, hisz ha egy programhoz nem mellékelik az összes videokártya-illesztőt, az már nem teljes értékű megoldás. Ahhoz, hogy a kompatibilitás látszata megmaradjon, az összes hardvergyártónak folyamatosan ügyelnie kell arra, hogy eszközei ne hagyjanak kezelési réseket a szoftverpalettán, de legalább a fontosabb programokkal szót kell érteniük.

Ebből a szempontból nagy könnyebbséget hoztak a grafikus felhasználói rendszerek (Windows, OS/2, X Window, BeOS stb.). Ezeknél csak a GUI-hoz kell videomeghajtó, mert (elvben) minden program a rendszermaghoz fordul perifériaigényeivel, így a problémák zöme megszűnik a jól dokumentált API-knak köszönhetően. Persze a valóság nem ilyen szép. Az API-kon alapuló programozás kényelmes, de a program lassabban fut, mintha közvetlenül macerálná valaki a „vasat”. Néhány alkalmazás számára a késedelem túl nagy ahhoz, hogy tolerálni tudja, ezért előáll az a helyzet, amikor az elvben kompatibilis alkatrészekből álló gépkombinációkon bizonyos szoftverek csak korlátozott teljesítménnyel futnak, vagy még úgy sem.

Jól megalkotott oprendszerek esetén a hardver minden elemét a kernelen át lehet elérni, így megvan a lehetőség arra, hogy ne a programok foglalkozzanak a milliónyi hardverelem apró kis eltéréseivel, hanem az operációs rendszer kezelje azokat. Tulajdonképpen ez a feladata. Csak az oprendszerhez kell elkészíteni a meghajtót...

A hardver kompatibilitásának másik bökkenője a fejlődés. A PC mai architektúrája immár második évtizedének végén jár. Az alapfelépítés még mindig az eredeti PC-t tükrözi. Pedig közben a technika elég sokat fejlődött. Bár az is igaz, hogy többnyire a sebességhajhászás volt a cél, nem pedig a rendszer ésszerűsítése. Gondolok itt a lehetetlenül kevés hardvermegszakításra (interrupt), a perifériák egymásba akadására és hasonlókra.

Ráadásul ezeket az ősi problémákat a „visszafelé kompatibilitás” jelszava ürügyén konzerválták. Közben a processzorok teljesítménynövekedésével egyre több régi program eleve használhatatlanná vált. Vagy azért, mert a processzor túl gyors lett, vagy mert megváltozott a belső architektúra. Elég, ha csak a PII óta jelentkező Turbo Pascal CRT Unit hibára gondolunk. (Ennek javítását egyébként a Borland szabadon elérhetővé tette.)

Szoftver

Ha gépeink összeálltak, többé-kevésbé megbízhatóan futtatják azokat az alkalmazásokat, amelyeket használni akarunk. Újabb kompatibilitási problémák keletkeznek azonban, amint össze

akarjuk kapcsolni ezeket a gépeket, vagy adatot akarunk eljuttatni egyikből a másikba. Ez utóbbi az egyszerűbb, mert az adattárolásnak van valamiféle szerkezete, az adathordozókat is elég régóta gyártják, így szinte minden gép képes elolvasni egy floppyt vagy CD-t.

A kisebb-nagyobb akadályok ellenére ezen a szinten ma már valóban elenyésző az ütközés. A „nagy floppyk" szinte teljesen kihaltak, csak a 3,5 hüvelykeseket használjuk, illetve nagyobb adatmennyiség vándoroltatására ott van a CD, az A-drive, a Jaz és a Zip lemez. E meghajtók vagy benne vannak a másik gépben, vagy nem, de hiányuk nem kompatibilitási hibára utal, hanem egyszerűen átgondolatlanságra az adatot mozgatni kívánó részéről. A nem általánosan elterjedt adattárakra nem lehet számítni...

Vannak persze formátumeltérések például a CD-k esetében, de ez általában nem az adatokat érinti, hanem például az állományok nevét (hosszú fájlnevek).

Az oprendszerek nem igazán képesek egymás fájlrendszereit kezelni. A Warp specialitása a HPFS, az NT-é az NTFS, a Windows a Win98-tól kezdve a FAT32-t használja, a Linux meg az EXT2FS-en szeret dolgozni. Ha több rendszerünk van egy gépen, és ezek között néha adatot kell cserélni, akkor gondunk támadhat. Az OS/2 nem kezeli az NTFS-t, a FAT32-t és az EXT2FS-t. A Windowsok sem ismerik a többiekét, de még egymásét sem (bár már NT alá már van a FAT32-nek kezelője). Érdekesség, hogy a 3.51-es Windows NT még gond nélkül kezelte a HPFS-t, az újabbaknál ehhez már trükköket kell csinálni. Pedig a fájlrendszer nem változott, sőt az NTFS eredetije a HPFS volt. A Linux-szal a legkönnyebb a dolog, mert olvasni mindegyiket tudja, írni viszont a HPFS-t és NTFS-t (ma még) nem. A vázolt feladatra egy FAT16 partíciót kell kreálnunk, ezt az ősi és elavult rendszert szerencsére mindenki jól ismeri.

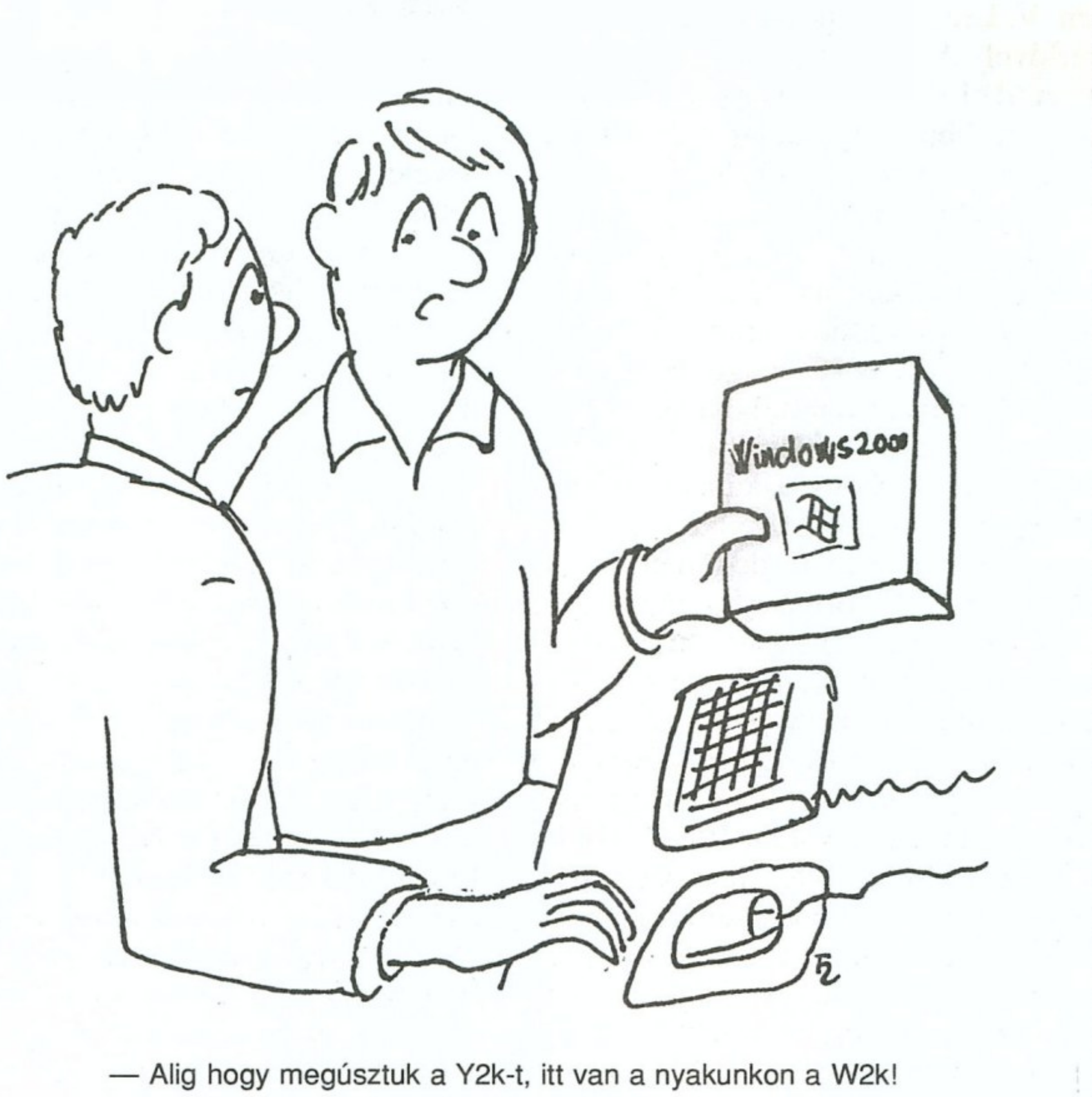
Problémát igazából maga az adat tud okozni. A különböző operációs rendszerek közötti adatcseréhez eleve fel kell készülnünk az eltérésekre. A szoftveriparban a nem teljesen korrekt verseny szülte az egymással szembeni szurkálódást. Az egyik rendszer szövegszerkesztője (vagy pontosabban fogalmazva dokumentumszerkesztője) szinte csak véletlenül képes olvasni egy másiknak a formátumát — akkor is, ha egyébként karakterkészletük egyforma. Ezt akár tudomásul is vehetjük, hisz azt senki nem ígérte, hogy egy OS/2-es program kezelni fogja a WinWord dokumentumait. Megtehetné persze, sőt egy-két éves lemaradással meg is teszi, csak közben a WinWord gyártója folyamatosan feladja a leckét a többieknek: verzióról verzióra változtatja a dokumentumok formátumát. Néha olyan apróságokkal kedveskednek felhasználóiknak, hogy az új verzióba „elfelejtik" beépíteni azt a programrészt, amely saját régebbi típusú dokumentumaik olvasását tenné lehetővé, és ezt utólag kell „a népnek" kikövetelnie. És ez még fokozható: a Microsoft Office azonos verziójú, de különböző nyelvű változatai nem képesek egymás makróit kezelni. Előállt tehát egy olyan jelenség, amely elvi megközelítésben lehetetlen: a program önmagával sem kompatibilis, mert ha egy angol Excel95 xls-t átviszek egy magyar Excel95-be, az nem tudja értelmezni a makrókat.

A dolog még szomorúbb, ha egy Word dokumentumot kiküldünk egy HP LaserJet nyomtatóra, majd ugyanabból a programból egy másik nyomtatótípust választunk. A két nyomat még akkor is eltér egymástól, ha egyébként nagyjából ugyanolyan a felbontás. A dokumentum a nyomtatásban válik véglegessé. Ha az egyik nyomtató 300 dpi-s, a másik 600 dpi-s, és emiatt újratördeli a lapokat, az már elég bizarr.

A kódlapok eltérései tovább nehezítik a helyzetet. Itt szintén nemcsak egyes rendszerek között, hanem néha az oprendszerek verziói között is óriási eltérések vannak. Gondoljunk például a Windows 95–98, pontosabban az Office 95–97 között elkövetett unikódosításra. Régi dokumentumainkat vagy újra kellett gépelni, vagy mindenféle trükköket kitalálni (konverziós programokkal, makrókkal), hogy meg lehessen menteni az egyszer már elvégzett munkát. Ez eltérő oprendszerek esetén sem valami lelkesítő, de valahogy még lenyeljük. Azonos rendszeren belül, azonos programcsomaggal már abszurd az egész.

Persze ilyen jelenségek nem csak a Microsoft jóvoltából fordulnak elő. A Red Hat Linux 6.1-es verziójától az X Window grafikus felületen futó programok is hasonló galádságokat művelnek ő és ű betűinkkel. Van, amelyik jól kezeli, van, amelyik nem. Itt viszont legalább a programok konzekvensek: egy bizonyos program minden verziója vagy jól csinálja, vagy rosszul. Ebben az esetben egyébként a gondok forrása a kétféle karakterelnevezés. Az ő és ű betűk korrekt neve tulajdonképpen „odoubleacute" és „udoubleacute", de ezek egy darabig nem voltak egységesen elfogadott kifejezések, és az ő-t

EZREDVÉGI KOMPATIBILITÁS



— Alig hogy megúsztuk a Y2k-t, itt van a nyakunkon a W2k!

„otilde", az ű-t „ucircumflex" néven kezelték, amihez az ISO 8859-2 kódlap megfelelő kinézetű karakterét rendelték hozzá. Van program, amelyik az egyik verziót érti, van, amelyik a másikat. Szerencsére egyetlen paranccsal válthatunk a két billentyűzetkiosztás között, de ez akkor sem szép dolog. Hozzáteszem: ha egy dokumentumban már benne vannak ezek a karakterek, akkor a programok jól mutatják és jól nyomtatják azokat, csak a billentyűzetről nem tudjuk megfelelően beírni, ha a program a másik térképet „érti".

Az egyes oprendszerek közötti kapcsolatteremtés során azt senki nem várja el, hogy egy Windowsra készült program fusson OS/2 vagy Linux alatt, de még BeOS-on sem! Ez visszafelé is igaz. Ha emulátorral megfejeljük a rendszert, vagy az OS/2-höz telepítjük a WIN-OS2-t, akkor persze bizonyos Windows programok képesek a másik rendszeren is működni, de a „tiszta" rendszer csak saját programjait tudja megfelelően futtatni.

Nyílt forráskódú szoftverek esetén viszonylag könnyű egy program átírása (portolása) egy másikra, de azok ilyen szinten nem lesznek „kompatibilisak" egymással.

A portolás egyszerű eljárás, ha rendelkezésre áll a program forrása, és a két rendszer közötti eltérések ismeretanyaga. A forrást módosítjuk (sok esetben nem is kell, mert a környezet mindkét rendszeren megvan, például az OS/2-re a GNU fejlesztő és futtató rendszerek, mint az EMX), és csak a célrendszeren kell újrafordítani a programot. Ezt akár kompatibilitásnak is felfoghatjuk, hisz ugyanaz a program két egymástól alaposan eltérő oprendszeren ugyanazt produkálja. Mégsem beszélünk kompatibilitásról, mert fordítás közben kerülnek a kész programba azok a részek, amelyek eltérőek. Továbbra sem lehet a kész bináris állományt egyszerűen csak átvinni, és a másikon használni.

Hálózat

Ha módunkban áll hálózatba kötni a gépeket, elég jól el tudnak egymással „beszélgetni". Persze megfelelő hálózati protokollt kell kialakítani. Minden rendszer érti a TCP/IP-t. Elvben az IPX-et is, de ezzel a protokollal mind a Windowsnak, mind a Linuxnak vannak kisebb bajai. A Novell NetWare (NW) szerverekhez a kapcsolódás fő protokollja az IPX. Az 5-ös NetWare óta ugyan IP-vel is mehet, de NW környezetben mégis az IPX a favorit. Minden rendszer tartalmaz klienst a NetWare szerverhez, és IPX-támogatást is. Maradéktalanul azonban csak az OS/2 tud saját IPX-hajtányával és a Novell kliensével dolgozni (meg természetesen a DOS, tisztán Novell megoldással). A Windowsok adnak saját IPX-et és NW klienst, mindkettő megbízhatatlanul működik, verziófüggetlenül (a Windows 2000-et még nem teszteltem).

Ha valahol ilyen rendszert kell üzemeltetni, első lépésként mindenképp érdemes a Novell által rendszeresen frissített NW klienst telepíteni. Ez az eredeti IPX csatolót is lecseréli egy jól működőre. Sajnos a NW kliens a bejelentkezési procedúrát észrevehetően lelassítja, de utána stabil kapcsolatot tart a szerverrel, ami az eredeti MS kliensről nem mondható el. A Linux NCPFS (gyakorlatilag ez a NW kliens) csomagja még fejlődőben van, az IPX néha „megszórja" a hálózatot felesleges csomagokkal. Bár használható, de a Microsoft kliensnél is kevésbé stabil. Érdekes, hogy egy régi, jól dokumentált szabványú protokoll meghajtóprogramjának programozásába eddig mind a Microsoft, mind a Linux közösség bicskája beletört. Az eltérés: a Linuxét még fejlesztik, a Microsoft úgy érzi, hogy az övé már kész...

A TCP/IP a Unix világból jött protokoll. Elvben itt sem lenne szabad, hogy a különböző rendszerek között eltérés legyen az IP stack megvalósításában, de kisebb gödrök bizony előfordulnak. Az egyik: a Windows kliensek által használt TCP/IP (amivel a megosztásokat képesek egymástól „elkérni" és használni) nem tiszta IP, hanem IP-be csomagolt Netbios. Emiatt néha kicsit furcsán viselkedik. Nem teljesen illeszkedik az IP-ben megszokott DNS-re, inkább a saját WINS névfeloldását részesíti előnyben, sokszor fölösleges csomagokkal árasztja el a hálót, és van még néhány kellemetlen szokása.

Az internet/intranet kapcsolatokban használatos TCP/IP szolgáltatási csomag viszont minden rendszerben egyforma. Végre egy jól felfogott közös érdek, hisz amelyik rendszer nem úgy kezel bizonyos csomagokat, ahogy az a szabványban van, az kizárja magát a nethasználók sorából. Az IP-t használó alkalmazásokban már bőven vannak eltérések is. „Fejlesztés, fejlődés!" felkiáltással a Microsoft igyekszik ugyanis minden elterjedt szabványt vagy szabványnak tekinthető rendszert (HTML, Java stb.) egy kicsit megcsavarni. No nem nagyon, csak éppen annyira, hogy alkalmatlanná tegye a többiek programjait a microsoftos „találmány" korrekt kezelésére.

Kisebb-nagyobb döccenőkön átvergődve a különböző rendszerek egymás között is képesek megosztani erőforrásaikat:

— A Warp File and Printer Services csomagja gond nélkül eléri az NT vagy a Win9x megosztásokat, akár domain, akár Workgroup szinten.

— Az NFS kliense a Unix és Linux rendszerek Network File System nevű megosztásrendszerére tud csatlakozni.

— A Windows kliensek elérik az Warp FaPS megosztásait, az eredeti csomagban nincs csatlakozó program a Linux szerverekhez (NFS kliens), de külön beszerezhető.

— A Linux a Samba klienssel képes mind a windowsos, mind a warpos megosztásokra belépni. A Samba szerver segítségével olyan megosztásokat tud kiajánlani a hálózatra, amelyekhez a Windows és az OS/2 saját hálózati klienseivel csatlakozhat.

— Természetesen a Linux NFS szerverhez a Warp illeszkedik, és a Linux kapcsolódhat a külön megvásárolható OS/2-es NFS szerverhez.

A Unix világában elterjedt használati mód: egy szerverre „betelnetelek", és ott futtatok programokat, amelyek (grafikus) képernyőkimenete és billentyűzet/egér bemenete a saját munkaállomásomra érkezik. Ez az X terminál rendszer. A munkaállomások viszonylag kicsik lehetnek, a szerveren egy helyen kell az adminisztrációt elvégezni, a jól kitalált jogosultsági rendszerrel a felhasználók területei, programjai biztonságosan megóvhatók házon belül, és a külső behatolókkal szemben is. Ehhez a rendszerhez nemcsak linuxos, hanem OS/2-es munkaállomások is csatlakozhatnak az ingyenes XFree86 rendszer segítségével, illetve a windowsos kliensek egy ott használható X emulátorral. Ugyanez működik a Warp felé is, de az OS/2-es XFree86 alá jóval kevesebb program készült, mint a Linuxhoz, így nem valószínű, hogy sokan élnek ezzel a lehetőséggel. Kellemes tulajdonsága, hogy ugyanezzel a metódussal minden Unix rendszerre (SCO, AIX, S/390, Sun stb.) lehet csatlakozni, így komoly gépteljesítményhez juthatunk kisebb értékű „terminállal" is.

Ebben a rendszerben gyakorlatilag a hálózati megvalósítást (TCP/IP protokoll stack) és az X szervert kell kompatibilissá tenni egymással. Magyarán: betartani a vonatkozó szabványokat. Viszont éppen ez az, ami nem mindig és nem mindenkinek sikerül, vagy amit nem mindenki akar...

Sándor Gábor
saga@matavnet.hu

Egy gépen nyolc rendszer

VMware 2.0 — PC a PC-ben, avagy pillantás a jövőbe

A közelmúltban jelent meg a minden idők legjobb PC-emulátorának kikiáltott VMware 2.0-s változata. A Windows NT-n, Windows 2000-en és Linuxon futtatható VMware 2.0 már 8 különböző, úgynevezett vendég (guest) operációs rendszer (DOS, Windows 3.1, Windows 95, Windows 98, Windows NT, Windows 2000, Linux és FreeBSD) futtatását teszi lehetővé. A béta verziókban még benne volt az OS/2 támogatása is, ezt azonban a végleges termékből sajnos kivették.

A VMware alapötlete nem új, hiszen korábban már más cégek is próbálkoztak PC-emulátorok fejlesztésével, ám eddig egyikük sem ért el átütő sikert. A VMware viszonylag széles körű népszerűsége több tényezővel is magyarázható. Az azonos kategóriájú termékek közül ennek a legjobb a hatásfoka, az emulált PC teljesítménye már nem marad el nagyságrendekkel az azt futtató (host) PC teljesítménye mögött. A siker másik összetevője a PC hardver teljesítmény/ár viszonyának folyamatos javulása. Manapság egyre többen rendelkeznek olyan teljesítményű gépekkel, amelyeken az emulált PC sebessége eléri vagy meghaladja a klasszikus Pentium processzorral felszerelt gépek sebességét. Mivel a ma használatos operációs rendszerek és szoftverek többsége egy ilyen gépen már elfogadható sebességgel működik, ezért a kedvező pillanatban piacra dobott VMware meglepően sokak gépén állta ki a használhatóság próbáját.

Mire jó a VMware?

A VMware sokkal több hasznos dologra alkalmas, mint azt az első pillanatban gondolnánk. Az egy gépen egymás mellett több operációs rendszert használók számára ideális alkalmazás, hiszen nem kell ismételt particionálással meg bootmenedzserekkel vesződni, sem pedig folyamatosan bootolni, ha az egyik rendszerről a másikra akarnak átváltani. Természetesen futtathatók mindazon alkalmazások is, amelyeket az éppen használt vendég operációs rendszer támogat. A VMware ezzel a felhasználó gépét igazi multiplatformos környezetté változtatja.

Az emulátor azok számára is jó megoldás, akik ugyan nem dolgoznak több operációs rendszeren, viszont időnként szeretnének kísérletezni nem megfelelően tesztelt, kétes eredetű, esetleg vírusfertőzött alkalmazásokkal, ám ezt gépükön eddig nem merték megkockáztatni. A vállalatok számára is komoly költségmegtakarítás lehet a VMware, hiszen egy erős PC-re telepített virtuális gépcsoporttal a cég megszabadul a komoly beruházást és folyamatos karbantartást igénylő tesztkörnyezet létesítésétől.

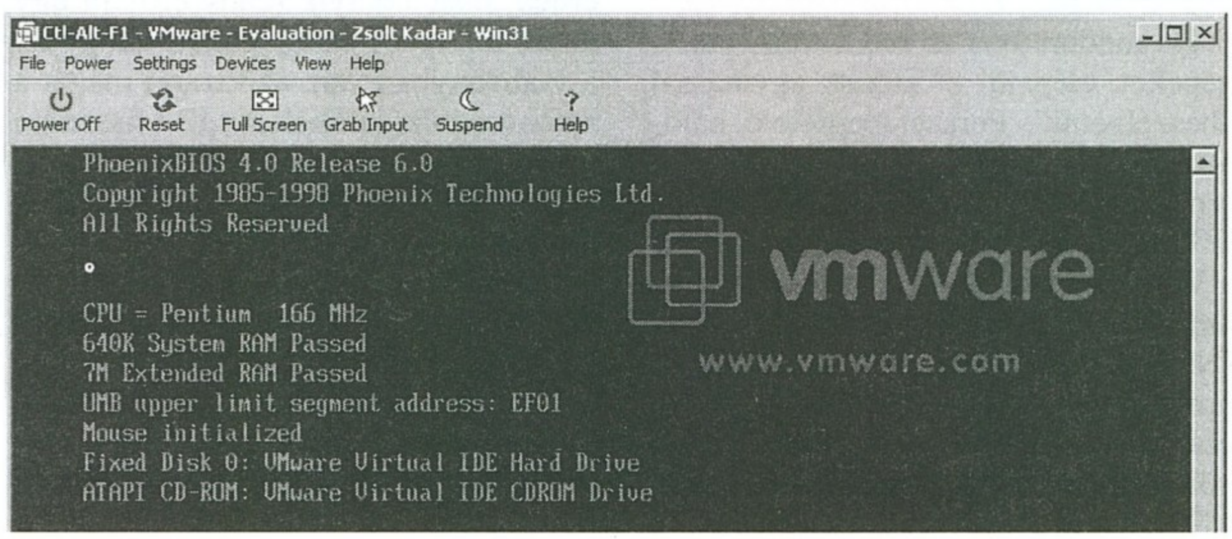
A VMware egyelőre persze még nem alkalmas minden elképzelt feladatra. A BeOS támogatása késik (hacsak az Other „operációs rendszer" nem tekinthető annak), és az OS/2 is csak a béta verzióban volt benne. A legújabb játékokat futtatók számára az emulált PC szinte biztosan lassú lesz, és a virtuális hálózati kártya korlátai miatt ez a változat a Token Ring hálózatokat használó vállalatoknál sem képes a hagyományos gépek kiváltására.

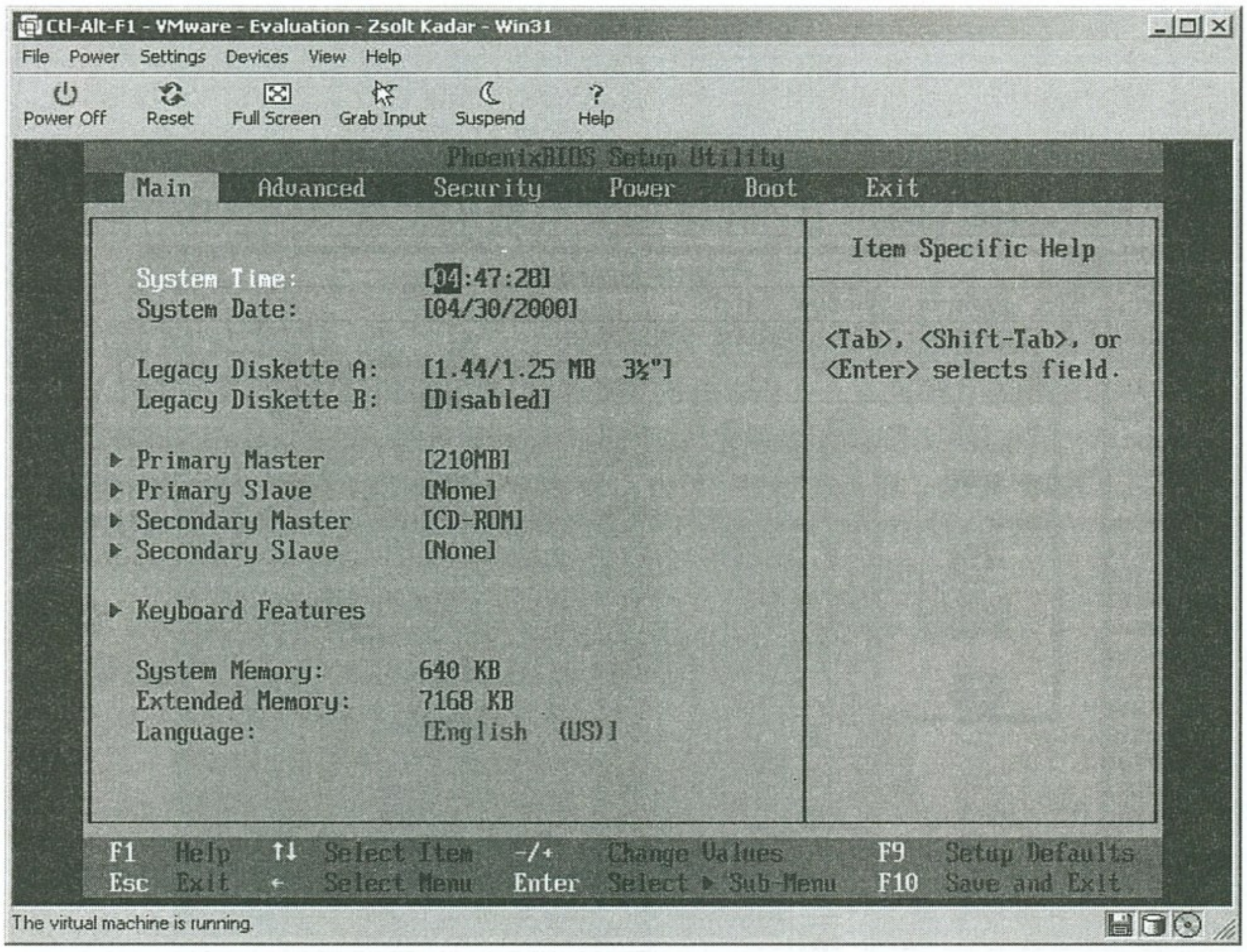
Telepítés

Ha úgy döntünk, hogy kipróbáljuk a VMware-t, akkor a következőket kell tenni. Első lépésként el kell döntenünk, hogy a gépünk alkalmas-e annak futtatására. A fejlesztő cég ajánlásai szerint legalább 96 MB RAM és PII-es, 266 MHz-es processzor kell hozzá. A tesztelés során én egy ennél lényegesen gyengébb (64 MB RAM, PI 166 MHz-es CPU) és egy lényegesen jobb (128 MB RAM, PIII 500 MHz CPU) PC-t használtam. Bár az alapötlet kipróbálására a P166-os is elég volt, mégsem ajánlom, hogy ilyen géppel bárki is belevágjon. A PIII-ason viszont már egészen gyorsan futott a Linux és az OS/2 is. Akár rendszeres használatra is alkalmas ez az összeállítás. A tapasztalat szerint nem is a processzor, hanem inkább a rendelkezésre álló RAM memória bővítésével érhető el a leglátványosabb teljesítményjavulás.

Amennyiben gépünk megfelel a specifikációnak, akkor második lépésként le kell töltenünk a http://www.vmware.com oldalról az alig több mint 5 MB-os próbaverziót, majd ugyanitt igényelni kell egy próbalicencet. Ez utóbbit emailben küldik el, 30 napos használatra jogosítva fel bennünket. Licencre mindenképpen szükségünk van, különben a virtuális PC-t telepítés után nem tudjuk bekapcsolni. Aki 30 napon túl is használni szeretné ezt az alkalmazást, annak bizony elég sokat kell érte fizetnie. A dobozos változat 329, a diákváltozat (vállalati környezetben nem használható!) 99 dollárba kerül. További kellemetlenség, hogy Windowsra és Linuxra külön (!) licencet kell venni, és az upgrade az 1.0-s verzióról sem olcsó mulatság.

Én a Windows 2000-re írt változatot teszteltem, amelynek installálása mindössze néhány percet vett igénybe, és csak néhány, viszonylag egyszerű kérdést kellett megválaszolnom.





Bállítási lehetőségek a CMOS-ban

Ha a sikeres telepítés után a program ikonjára kattintunk, akkor megjelenik egy dialógusablak, benne automatikusan ki van választva az a varázsló, amelynek segítségével megtervezhetjük az első virtuális gépet. A varázslóban lehet megadni, hogy milyen operációs rendszert szeretnénk futtatni, és hogy mekkora merevlemez-területet akarunk a virtuális gép rendelkezésére bocsátani. Az alkalmazás ezen részét később is lefuttathatjuk a File menüből, és ezzel újabb virtuális gépeket hozhatunk létre. A konfigurációk között kedvünkre váltogathatunk, vagy egymás mellett is futtathatjuk őket.

Bekapcsoljuk a virtuális gépet

A VMware által emulált gépet — a valódi számítógépek mintájára — használat előtt „be kell kapcsolni". Ekkor bejelentkezik a Phoenix típusú BIOS, és lefutnak az ilyenkor szokásos tesztek. Első indításról lévén szó a virtuális gép az „Operating system missing" (hiányzik az operációs rendszer) üzenettel örvendezteti meg használóját. Ez így is van rendjén, hiszen még nem telepítettünk semmit a virtuális gép virtuális merevlemezére. (A VMware képes meglévő partíciókra telepített operációs rendszerek elindítására is, ám ezt a működési módot csak a tapasztalt felhasználóknak ajánlják.)

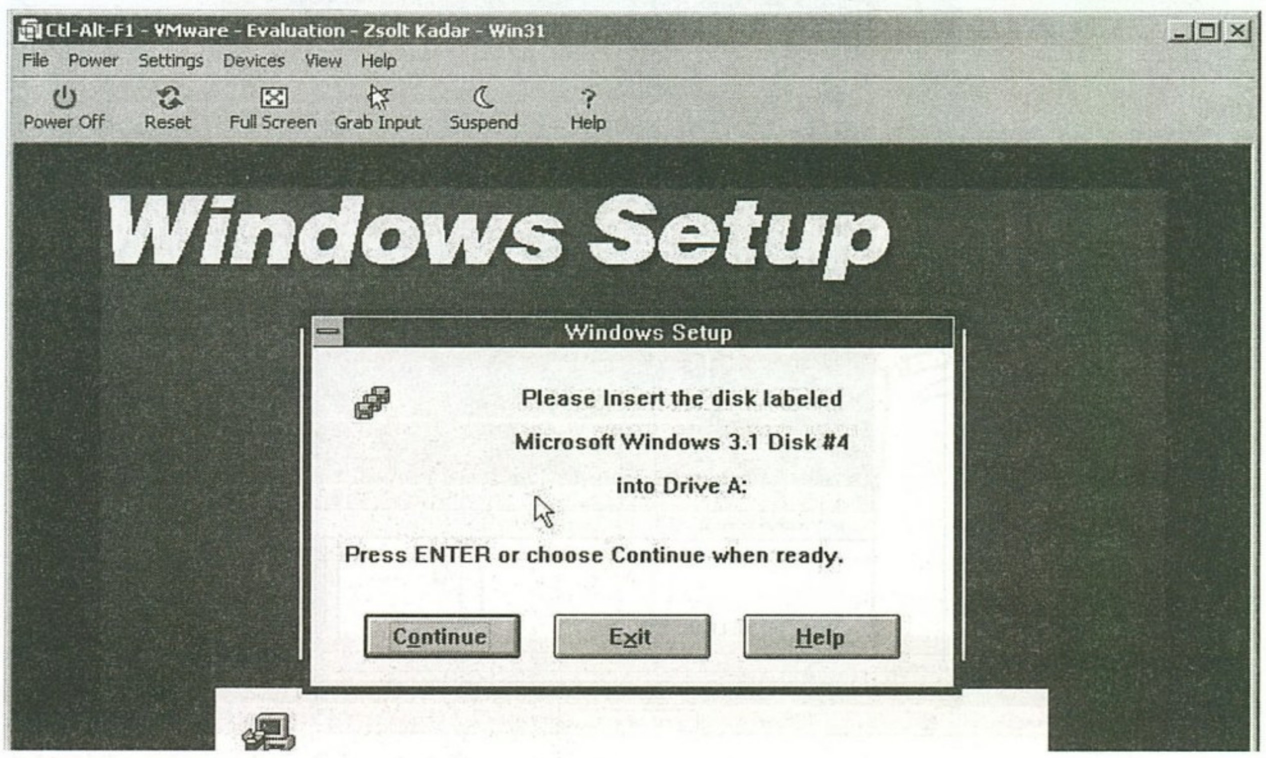
Virtuális gépünkre tehát telepíteni kell egy operációs rendszert és azon futtatandó alkalmazásokat. A VMware virtuális floppyval, IDE CD-ROM-mal és Ethernet hálózati kártyával is ellát bennünket. A guest operációs rendszer telepítése elvileg tehát történhet floppyról, CD-ről vagy a hálózatról is. Én a DOS, Windows 3.1, Linux, valamint az OS/2 telepítésével próbálkoztam (ez utóbbival a 444-es béta verzió alatt), s az minden esetben problémamentesen lezajlott.

A VMware MMX processzort emulál. Többprocesszoros gépek esetén dedikált CPU-kat is lehet rendelni az egyes virtuális PC-khez. Sajnos 3D-s grafikus eszközöket, valamint MIDI eszközöket még nem képes virtualizálni. A 2.0-s verzióban a teljesítményen kívül lényegesen javítottak több más eszköz (CD-ROM, SCSI, soros port stb.) támogatásán is. Külön érdekesség még, hogy a floppy és a CD-ROM menet közben le- és felcsatolható, ami jelentősen javítja a rendszer használhatóságát.

Használat közben

Amikor szerencsésen túljutottunk az operációs rendszer és a hozzá tartozó alkalmazások telepítésén, fordítsunk egy kis figyelmet a virtuális PC finomhangolására. A legelső ilyen hely — csakúgy, mint egy valódi PC esetében — a CMOS Setup. A bootolás elején megnyomva az F2 billentyűt, belépünk a CMOS beállítások menüjébe, ahol érdemes ellenőrizni, hogy a VMware által megadott alapbeállítások megfelelnek-e elképzeléseinknek. A második beállítási pont az emulátor Global Settings menüje, ahol például olyan dolgokat szabhatunk meg, hogy a rendelkezésre álló fizikai memóriából mennyi memóriát foglaljon le a VMware az emulátor számára, mennyi memória álljon az egyes virtuális PC-k rendelkezésére, vagy milyen prioritások legyenek érvényben a háttérben, illetve az előtérben történő futtatáskor. A globális beállításokat felülbírálják a virtuális környezetenként megadott lokális beállítások, melyeket a Settings főmenüből lehet elérni.

A finomhangoláson kívül az ún. VMware Tools (VMware kiegészítő eszközök) telepítésével is lehet fokozni az emulátor teljesítményét. A 2.0-s verzióba már integrálták ezt a kiegészítést, amelyet ugyancsak a Settings főmenüből lehet telepíteni. (Az OS/2-t is támogató bétához — csakúgy mint az 1.0-s verziókhoz — a kiegészítést még külön kellett letölteni, és floppyról telepíteni.)



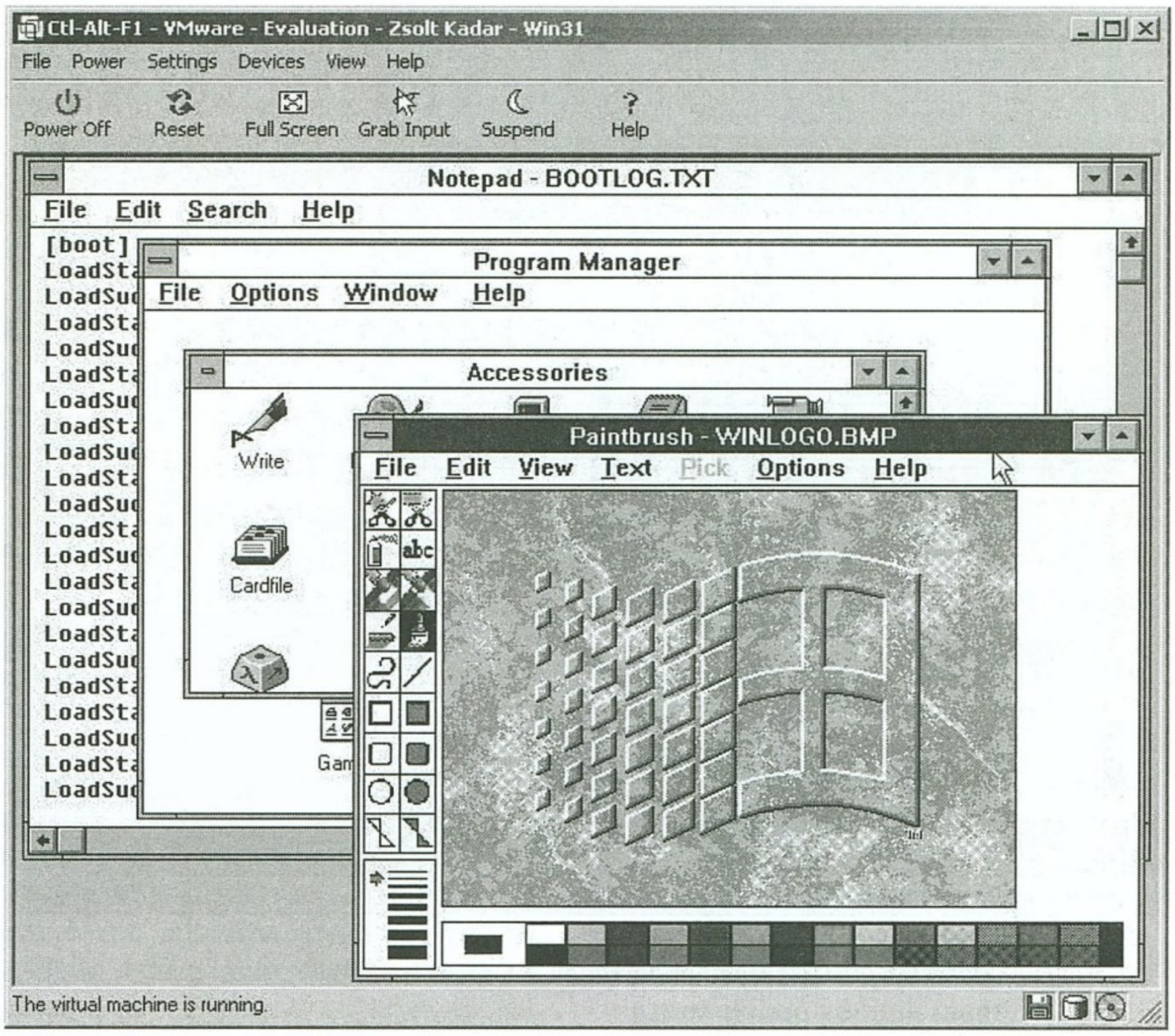
Windows 3.1 telepítése a virtuális PC-re

A VMware Tools felrakása után virtuális PC-nk képes lesz az SVGA üzemmódok támogatásra, a guest rendszer órájának a hostéhoz történő igazítására, valamint a host és a guest közötti, vágólapon keresztül történő adatcserére (cut & paste).

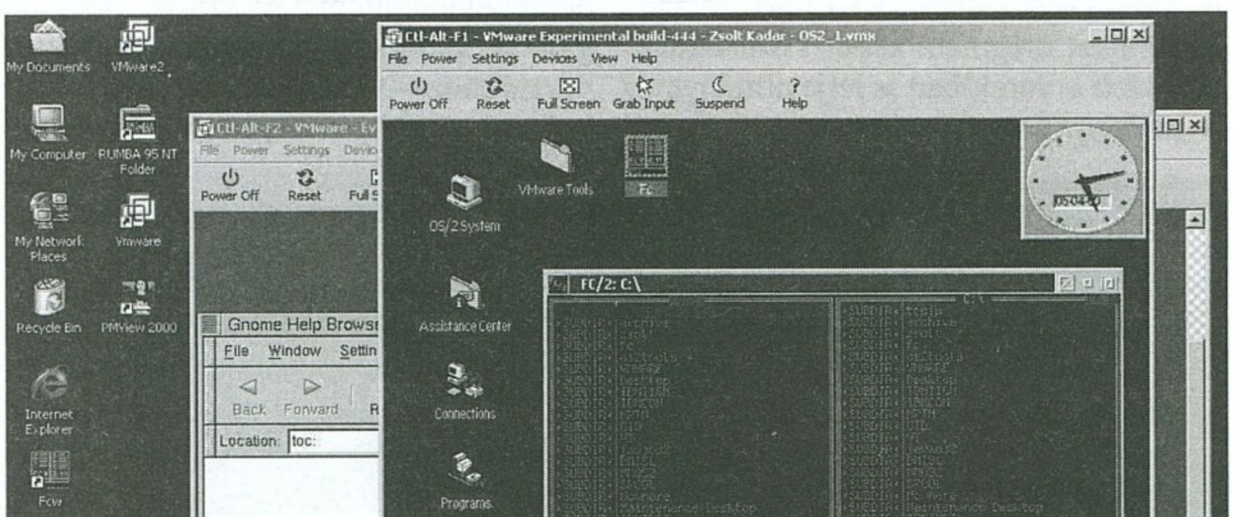
Ha futtatás közben bármilyen problémánk lenne a virtuális géppel, akkor azt a VMware-ben legegyszerűbben a Ctrl+Alt+Del-nek megfelelő Ctrl+Alt +Backspace billentyűkombinációval indíthatjuk újra. Ha ez nem segít, akkor megnyomhatjuk a virtuális PC Reset gombját is, amely — csakúgy, mint a valódi PC-k esetében — garantáltan működik. A szimuláció annyira tökéletes, hogy az újraindított operációs rendszer ilyenkor észleli, hogy a számítógép nem a megszokott módon lett kikapcsolva, ezért végrehajtja a merevlemez szokásos ellenőrzését.

Különösen a lassan induló operációs rendszereknél lehet hasznos az új hibernálási (suspend) opció, amelynek segítségével elmenthetjük a futtatott rendszert annak egy nyugodt pillanatában, és ahelyett, hogy egy rendszerösszeomlás után végigvárnánk a hosszadalmas lemezellenőrzést és a bootolást, egyszerűen visszatöltjük a hibernált állapotot.

Aki más rendszerekkel is kommunikálni akar, annak engedélyezni és konfigurálni kell a VMware által virtualizált, AMD Ethernet Family Adapter típusú hálózati kártyát. Ezt szinte mindegyik operációs rendszer támogatja, így nem probléma a megfelelő driver beszerzése (amennyiben a telepített rendszer nem ismerné fel a kártyát automatikusan). Az OS/2-es béta verzióban sajnos elég gyengére sikerült a virtuális hálókártya, ugyanis csak a TCP/IP protokoll működik vele, ami eléggé megnehezíti NETBEUI (Netbios) és TCPBEUI (Netbios over TCP/IP) protokollokon alapuló adatcserét a windowsos gépekkel. A végleges változatban támogatott operációs rendszerek hálózatkezelésére szerencsére nincs panasz. A linuxos VMware telepítésekor például automatikusan felkerül a Samba is, ezáltal nagyon egyszerűvé válik a Linux host és a Windows guest közötti adatcsere.



Ismerős, ugye?



OS/2, Linux, s mindez Windows 2000 alatt futtatva

Végkövetkeztetések

A VMware 2.0 a maga nemében egyedülálló program. Megfelelően erős gépen lehetővé teszi a manapság legszélesebb körben használt operációs rendszerek és az azokon futtatható alkalmazások használatát, s ehhez nincs szükség merevlemezünk partíciókra tördelésére, bootmenedzserek telepítésére, vagy az egyik rendszerből a másikba történő váltáskor újbóli bootolásra.

Habár az emulált PC felhasználóinak még jó néhány korláttal kell számolniuk, sokak számára már ma is ideális megoldás lehet a VMware. A magyar átlagkeresettel rendelkezőknek inkább a termék meglehetősen borsos ára jelenti az igazi elriasztó erőt. Ennek ellenére csak javasolni tudom mindenkinek, hogy ha van lehetősége rá, próbálja ki ezt a nagyon jó minőségű, az operációs rendszerek és az alkalmazások eddigi eredményeit messzemenően hasznosítani tudó, a kompatibilitás jövőbeli megvalósításának lehetőségét hordozó alkalmazást.

Kádár Zsolt
kadzsol@xenia.sote.hu

„Van egy kis probléma...”

Számítógép-bővítési fronton a helyzet változatlan?

Mielőtt hozzáértő vállalati olvasóink belemélyednének a nagy rendszerek kompatibilitását taglaló — a következő oldalakon elhelyezett — írásokba, helyezzék magukat kényelembe, és tegyenek egy kis kirándulást a „low end” környékén, főleg ha helyzetükből adódóan személyesen egy ideje már nem találkoztak ilyen problémákkal. Az itt következő kis „esettanulmányban” egyik külső munkatársunk írta meg a vele történteket.

Lassan már 3 éve nyúzok egy 486-os gépet. Illene korszerűsíteni egy kicsit, hogy hardverileg is haladjak a programok fejlődésével. Tehát gépújításra adtam a fejem. Milyen egyszerű: alaplap, proci, vezérlőkártya... és már meg is van. Kár lenne kidobni a többit. A hangkártya, a modem, a videokártya, a winchester, a floppydrive stb. jól működik, sőt néhány kártyát nemrégiben vettem.

1

Bevonszolom a gépet az üzletbe, melyben 66 MHz-es, 486-os DX2 proci van az alaplapon. Előadom, hogy szeretném kicseréltetni Pentium II alaplapra, 233 MHz-es procival. Ajánlgatják a 300-as Celeront, de amikor már-már meggyőztek, közlik az árát is. Inkább maradok a 233-asnál. Egyből küldik a szervizbe, és megnyugtatnak, hogy amikor délután visszamegyek, már vihetem is haza.

2

Délután — zárás előtt — közölték, hogy minden rendben, csak van egy kis probléma. Ekkor eszembe jutott a kálváriám, amikor annak idején a 386-osomat cseréltem 486-osra, és rosszat sejtve kérdeztem, hogy mi a gond. Nincs nagy baj, csupán annyi, hogy túl sok „régi” VESA kártyám van, melyeket nem lehet rárakni a PCI buszos alaplapra, mert összevissza csak két VESA hely van rajta.

Javasolták, hogy kicserélik a videokártyát, és ha beveszik tőlem a régit, már csak 3000-et kell ráfizetnem. A hangkártyát is kicserélik szívesen, mert van raktáron PCI-s Sound Blaster. Amikor rákérdeztem, hogy az működik-e majd így DOS alatt, kissé furcsán néztek rám. Mi a fenét akarok én? Felvilágosítottam őket, hogy én már csak ilyen őskövület maradok, s valahogy nem igazán venném jó néven, ha csak Windows alatt működő hardverelemekkel lennék gazdagabb, 5000 forinttal pedig szegényebb, mivel erre is rá kellene fizetni.

Akkor már éreztem, hogy erre tényleg jól rá fogok fizetni, pedig csak utána jött a feketeleves. Na, rendben van, a hangkártyának megkegyelmeznek, ha DOS-t akarok, viszont akkor a belső modemet kell lecserélni egy külsőre. Így már drágább: 10 ezer forinttal többbe kerül a buli, de ez is csak azért, mert „jó áron” átveszik tőlem a régi belsőt.

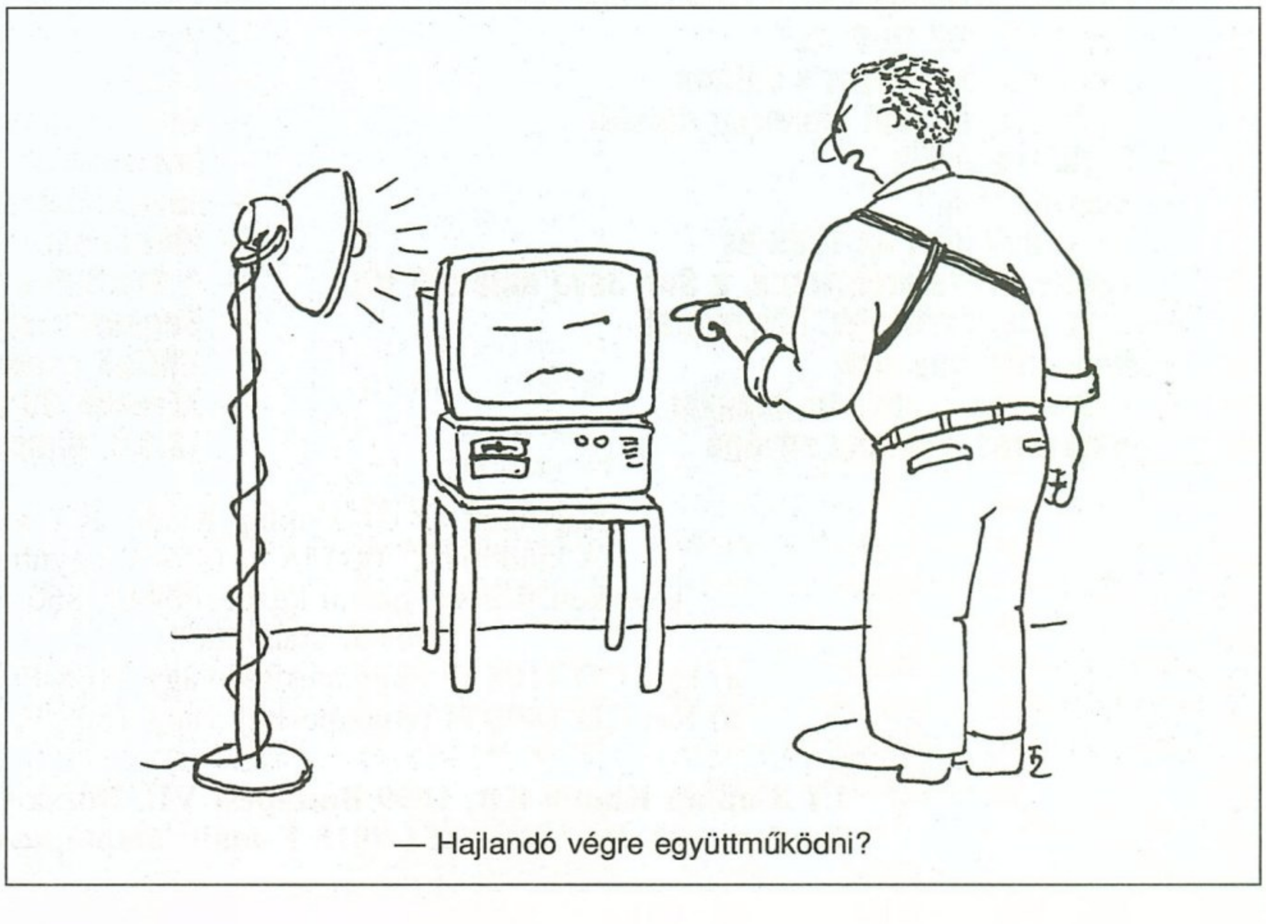
Hogy ne húzzuk tovább a dolgot, beleegyeztem. Fél perc múlva ismét egy probléma: jelenleg nincs külső modem. Mi legyen? Lehet, hogy lesz a jövő héten. Vagy ki tudja. Addig mit csináljak? — kérdeztem. Használható ötletük nem nagyon volt. Ekkor feltűnt nekik, hogy nekem két printerkártyám van. Mi a fenének ez? Felvilágosítottam őket, hogy ez kell, ugyanis nemcsak a nyomtató kötendő párhuzamos portra, hanem más is, és erre mindenképpen szükségem van. Félénken kérdeztem, hogy esetleg van PCI-s párhuzamos portjuk...

Nincs! Végül megegyeztünk abban, hogy próbálnak szerezni egy külső modemet, és akkor nem kell cserélgetni.

3

Másnap az üzletben újabb probléma fogadott. Az a nyamvadt szkennerkártya. Megkérdezték, hogy mikor vettem ezt a „régi” szkennert? Elárultam nekik, hogy kb. 2 hónapja, és éppen tőlük, új hardvereszköz gyanánt. Sebaj — mentegetőztek —, ez már régi, ebből is PCI buszos lenne jó, de akkor az egész szkennert ki kellene cserélni. Legegyszerűbb, ha rögtön SCSI-ra upgradeelek.

Ismét felötlött bennem, hogy én programozónak tanulok az egyetemen, de nekünk csak függvényábrázolást, meg elméleti matematikát tanítanak. Olyasmit, hogy jól gondoljuk meg, milyen alaplapot veszünk, azt bizony nem. De a RISC processzor lábkiosztásait a zéhára tudjuk, mint annak a rendje. Felajánlották, hogy keresnek egy olyan



— Hajlandó végre együttműködni?

alaplapot (nem biztos, hogy van náluk ilyen, mert baromi ritka manapság), amelyen nem 2, hanem 3 VESA busz van, és így talán megúszható a szkennercsere (ára 50 ezer felett). Hurrá, minő szerencse, van alaplap 3 VESA-val, és mindössze 7000 forinttal kell többet fizetni. Gyors számolgatás, spekulálás, egye fene, legalább megoldódik ez a probléma. Végül arra jutottunk, hogy a drágább alaplapra fellökik a kártyákat, és holnapra készen lesz.

4

Másnap az üzletben már majdnem úgy nézett ki, hogy vihetem a gépet. Kiderült azonban, hogy így sem fér bele minden kártya, a videovezérlőt ki kellett cserélni. Apropó, milyen monitora is van? Még a régi szép időkből való Mono VGA. Húha, ez problémás. Érdemesebb lenne színest hozzákötni, mert nyafogni fog, ha nem az van rajta. Csekély összeggel többet kell fizetni. A régi monitort esetleg hajlandók bevenni... ha úgy gondolom.

A monitorpróba során kiderült, hogy a BIOS rá sem hederít, ha nem színes megjelenítő van a géphez kötve, és simán átállítja monokróm üzemmódba a videokártyát. Maradtam tehát az olcsóbb megoldásnál.

5

Hazavittem a gépet, miután 15 000 forinttal többet kifizettem érte. Nekiültem szoftveresen (is) rendbe rakni. No ezt rosszkor tettem, mert elment az étvágyam, pedig éhes voltam. Ami csak lehetett, az összeakadt: a szkenner a hangkártyával, a modem az egérrel. A CD-meghajtó nem mindig akarta olvasni a lemezeket...

Még jó, hogy legalább van egy tök jó gépem!

Miután szórakoztam egy keveset a telefonszámlám terhére (pedig csak a leveleimet szerettem volna letölteni), feladtam a modemes kísérleteket. A hangkártyám életre kelt, miután átdzsampereltem. A szkenner azonban nem akart működni, és a megmaradt egyetlen printerportra kötött nyomtató sem akart megmoccanni.

6

Az üzletben ismét elkezdték pedzegetni: „csak egy kicsit többe kerül...".

Végül mi lett az eredmény?

A CD-ROM kontakthibás vezetékének problémájára rájöttek, és „ingyen" megcsinálták, a többi hibát pedig a haverokkal közösen megoldottuk, két hétig bütykölve.

* * *

Ha belegondolok, hogy mennyire megszívattak, pedig nem is tekinthetem magamat teljesen laikusnak, akkor hogyan járhatnak azok, akikre esetleg hatással van a reklámkampányok új irányzata, hogy „számítógépet minden hozzá nem értő asztalára"?

Az eladott árucikkekhez (hardver) egysornyi leírást nem adnak, az egymással nem kompatibilis eszközök káoszában maguk az értékesítésben dolgozók sem tudnak eligazodni, és tévedéseiket a felhasználókkal fizettetik meg... pénzben, időben, idegeskedésben.

Mostanában a szakma nagy mágusai szeretik a számítástechnikai produktumokat az autóiparéhoz hasonlítani, a felhasználóktól elvárható hozzáértést pedig az „úrvezetők" szintjére helyezni. Kívánom nekik, hogy autójuk használatához ugyanolyan mértékben kapják meg a megbízhatóságot, a csereszabatosságot, a márkák közötti választási lehetőséget, a minőségi garanciákat, amilyen mértékben a számítástechnikai ipar és a kereskedelem azt jelenleg az alvégen szolgáltatja.

Pál Zsolt
pille@elender.hu

A vegyes környezet integrátora

SCO UnixWare 7 — és a többiek

A Unix szempontjából a windowsos hálózatokkal való integráció az egyik legfontosabb gyakorlati feladat. A kisebb hálózatok hasznosíthatják a Unix platformokon (Sun, HP, IBM stb.) futó SCO VisionFS programcsomagot, amely lehetővé teszi, hogy a Windowst futtató PC-kről a felhasználók transparensen érjék el a unixos fájlokat és nyomtatókat, tehát azok windowsos hálózati meghajtóként és nyomtatóként látszódjanak.

A Unix rendszerben „ablakot nyitó" VisionFS programcsomagot kizárólag a szerverre kell telepíteni. Segítségével a hálózaton minden Windows PC hozzáférhet az összes Unix fájlhoz és printerhez. Az NFS-sel ellentétben nem szükséges a szoftvert minden egyes PC-re telepíteni és konfigurálni. A munkaállomásokon Windows 95/98, Windows NT, NT, Win2k, Windows 3.x, OS/2 és DOS lehet. A felhasználók azonosítása történhet unixos nevük alapján, de lehet saját jelszófájlt is kezelni VisionFS alatt, sőt az NT-s domain kontroller ugyancsak használható.

A VisionFS lehetővé teszi, hogy a munkacsoportok az interneten keresztül is megosszák az erőforrásokat, és távoli Unix szervereket úgy kezeljenek, mintha azok a helyi hálózaton lennének. A termékben lévő Profil szerkesztőprogrammal egyszerűen és gyorsan lehet a hozzáférési jogokat szabályozni. A fontosabb paraméterek parancssorból is konfigurálhatók.

Akinek ennél több kell (például Windows NT szerverekkel való szoros integráció), annak az Advanced File and Print Server csomag kiváló megoldást biztosít. Ez az előbbieken felül domain kontrollerként is tud működni. Mivel az AFPS a Microsoft NT hálózati technológiáját használja, ezért azonos képességű gépek egyenrangú kapcsolatában (peer-to-peer) kompatibilis azzal, így a kliensek számára pontosan olyannak látszik, mint egy 4.0-s NT szerver. Megvalósult benne a WINS (Windows Internet Name Services) és a CIFS (Common Intenet File System) is. Kompatibilis az NT Server Tools-szal, ami lehetővé teszi a UnixWare 7-en található osztott erőforrások adminisztrációját windowsos gépekről is. A rendszer mind a felhasználó, mind pedig az adminisztrátor szemszögéből pontosan olyan, mint egy NT környezet.

Hasznos lehet a Merge nevű termék is, amely DOS és Windows alkalmazásokat képes futtatni. A közös fájlrendszer lehetővé teszi, hogy a Unix és DOS alkalmazások, illetve felhasználók megosszák adataikat, nyomtatóikat és egyéb PC-s perifériáikat. Támogatja a Winsock 1.1 hálózati szabványt a 16 és 32 bites alkalmazások számára.

A többi Unix rendszerhez való kapcsolódás természetesen eleve nem jelenthet gondot, hiszen a régóta bevált szabványok (TCP/IP, NFS, PPP stb.) alapján a Unix rendszerek jól megértik egymást, az integráció gyakorlatilag zökkenőmentesnek mondható.

A Unixware2 erős Novell alapjaira építve a Unixware7 opcionális terméke a NetWare Services (NWS), amely tartalmazza a NetWare fájlkezelési, nyomtatási és könyvtári (NDS) szolgáltatásait. Ezek a 4.10a verziónak megfelelő szolgáltatásokat nyújtják mind a kliens, mind a szerver oldalán. A rendszer teljes mértékben támogatja a Novell Directory szolgáltatásait és az API-kat, valamint a Novell Cross Platform Services-t. Grafikus adminisztrációs programok léteznek a telepítéshez és a konfiguráláshoz (például: NDS adminisztrálás, NetWare és Unix nyomtatók, kötetek, hálózati konfigurálás és hangolás). Egyaránt kezelni tudja a UnixWare-es és az NFS-sel mountolt fájlrendszereket. A felhasználók, az adminisztrátorok és a hálózatmenedzselő programok (például Novell ManageWise) számára teljesen átlátható. A Netware szolgáltatások IP kapcsolaton keresztül is elérhetők a UnixWare IPX tunnelling segítségével.

A unixos fájlok és nyomtatók DOS alóli elérhetőségét biztosító termékek nagy többsége NFS protokollt használ, amelynek szerveroldali része minden Unixban — így a UnixWare7-ben is — megtalálható.

A Unixware7 a szokásos Windows/DOS kliensek mellett — jó Unixhoz híven — széles körű támogatást nyújt más eszközökhöz. Ilyenek például a nagy reményeket keltő NC (network computer) gépek, vagyis az ún. vékony kliensek, a Java-képes „webtop" eszközök, a jól bevált karakteres terminálok (például VT220 stb.) és különböző ASCII perifériák. Az adott alkalmazás igényeinek legjobban megfelelő környezetet lehet mindenhol alkalmazni. Kiemelendő az SCO „Tarantella" technológiája, amely Java-képes böngésző segítségével (tehát gyakorlatilag semmiféle karbantartást nem igénylő kliens használatával) lehetővé teszi a hozzáférést bármilyen Unix, NT, mainframe, grafikus vagy karakteres alkalmazáshoz azáltal, hogy az adott alkalmazás felületét röptében „webesíti".

A hálózati számítástechnikához természetesen minden lehetőséget megad a Unixware7, többféle protokollt támogat, a legfrissebb internetes, Java, üzenetkezelő és címtárkezelő szolgáltatásokkal együtt. Természetesen támogatja a TCP/IP legújabb verzióját, az IPV6-ot és az ehhez kapcsolódó összes jelentősebb szolgáltatást, például NFS, NIS, NTP, DNS, DHCP, LDAP és még hosszan sorolhatnánk a 3-4 betűs rövidítéseket. A szabványos SNMP támogatásával a UnixWare integrálható bármilyen hálózatfelügyeleti rendszerbe, és mind ügynöki, mind felügyeleti állomás funkciót elláthat.

A kliens-szerver alapú mentőprogramoknak kliens és szerver változata egyaránt elérhető a Unix gépeken. Így könnyen tudunk Novell, NT, Windows gépekről Unix szerverre menteni. De az sem gond, ha Unix fájlokat kell NetWare-es gépre lementeni. Ez különösen a nagy, heterogén hálózatoknál fontos tulajdonság. A UnixWare 7 legtöbb kiadásában megtalálható az ARCserve IT mentőprogram. A nagygépes kapcsolatokban pedig általában ugyanazok a protokollok használatosak, mint a többi unixos géphez.

Pongrácz Tibor
tibi@areco.hu

NDS, az integráló erő

Heterogén hálózatok hatékony kezelése

A vállalatok számítástechnikai rendszere szükségszerűen egyre bonyolultabb és heterogénebb. Ezekben a kompatibilitás azt jelenti, hogy a sokféle eszköz és program képes egymással együttműködni. A cél azonban ennél is több: az együttdolgozásnak hatékonynak kell lennie, hogy jól működő informatikai rendszert kapjunk.

Minden együttműködés magvalósításának legjobb kiindulópontja az, hogy a rendszerfejlesztésnél figyelembe vesszük az adott terület uralkodó szabványait. A hálózatokban ennek alapfeltétele a közös protokollok használata.

A Novell alapú hálózatok, illetve azok „motorjai", a NetWare szerverek támogatják a legfontosabb hálózati protokollokat. Közülük az IPX/SPX magától értetődő, hiszen ez a lokális hálózatok számára a Novell által kifejlesztett szabvány. Az internet terjedésével a TCP/IP és az ehhez kapcsolódó szolgáltatásokat megvalósító protokollok szerepe azonban egyre nagyobb. A Novell ezért a NetWare 5-től kezdve lehetővé tette a TCP/IP-nek mint alapprotokollnak a használatát, ezáltal a hatékony együttműködés minden IP-s eszközzel megvalósítható.

A sokféleség adottság

A hálózatok működtetésének kulcskérdése a felhasználói munkahelyek támogatása. Az informatikai rendszerek fejlődésével egyre újabb, és egyre többet tudó munkaállomások készülnek, és ezek támogatása kritikus pont a hálózatok üzemeltetésében.

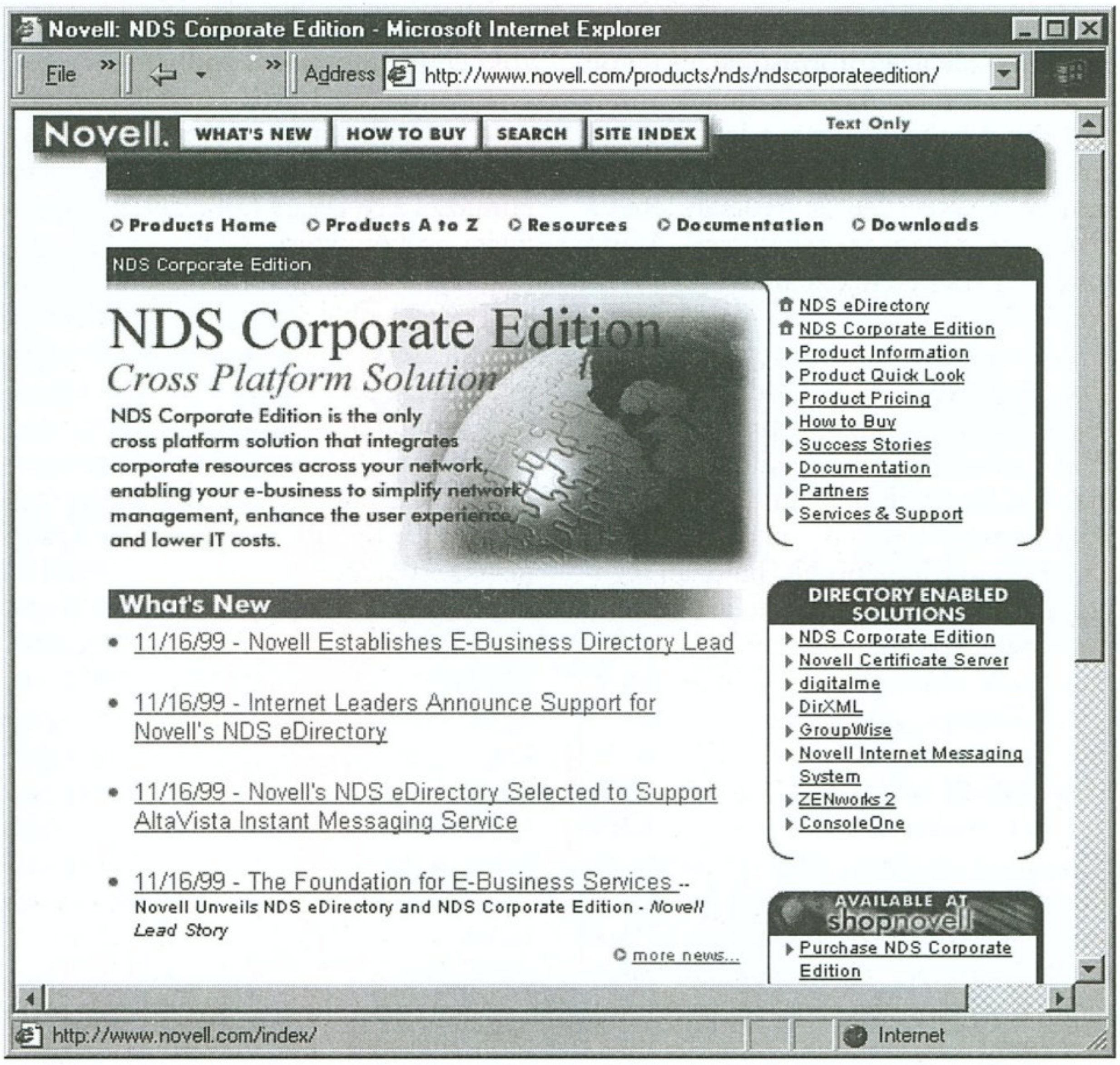
Egy NetWare hálózat erőforrásaihoz a legkülönfélébb desktop eszközökkel lehet hozzáférni. Létezik NetWare kliensprogram OS/2, Macintosh, Unix, DOS/Windows, Windows 95/98, Windows NT és Windows 2000 platformokon egyaránt. A magyar piacon különösen fontos, hogy akár egy kisebb teljesítményű, akár boot-epromos, merevlemez nélküli gépről is elérhetők legyenek a NetWare szerver szolgáltatásai. A másik fontos elem a böngésző alapú rendszerek támogatása. A böngészőprogramok gyakorlatilag minden platformon elérhetők, ezáltal a régi és az új hardvereszközök közel azonos eséllyel férnek hozzá az alkalmazásokhoz. A Novell termékei esetében több programhoz is létezik ilyen interfész, például a GroupWise WebAccess révén elérhetők a „postahivatalunkban" tárolt információk, a NetWare 5.1 Management Portal modulja segítségével pedig szervereink felügyeletét végezhetjük el nagyon egyszerűen.

A hálózati alkalmazások közül mindig olyanokat célszerű választani, amelyek illeszkednek a meglévő informatikai rendszerhez. A PC-s hálózatokban az első nagy generációt a Clipper alapú programok jelentették, míg az elmúlt néhány évben a fejlesztést a Win32-es kliens-szerver alapú alkalmazások uralták. A tendencia jelenleg egyre inkább a Java irányába mutat, melynek terjedését két dolog segíti. Az egyik az internet, hiszen a Java az e-business világában egyre inkább meghatározó komponens. A másik éppen a kompatibilitás, lévén a Java platformfüggetlen, így a programok változtatás nélkül (vagy minimális módosítással) átvihetők egyik operációs rendszerről a másikra.

A NetWare felügyeleti programja ConsoleOne Java alapú, így a hagyományos Win32-es platform mellett „otthon van" Solarison és Linuxon is. Másik lényeges komponens a NetWare 5.1-ben megtalálható a WebSphere, az IBM Java alapú alkalmazásszerver környezete és első számú e-business platformja. A Javának köszönhetően a NetWare 5.1 megjelenésével minden WebSphere alkalmazás elérhetővé vált ezen a platformon is, függetlenül attól, hogy addig AIX vagy AS400 alatt futott.

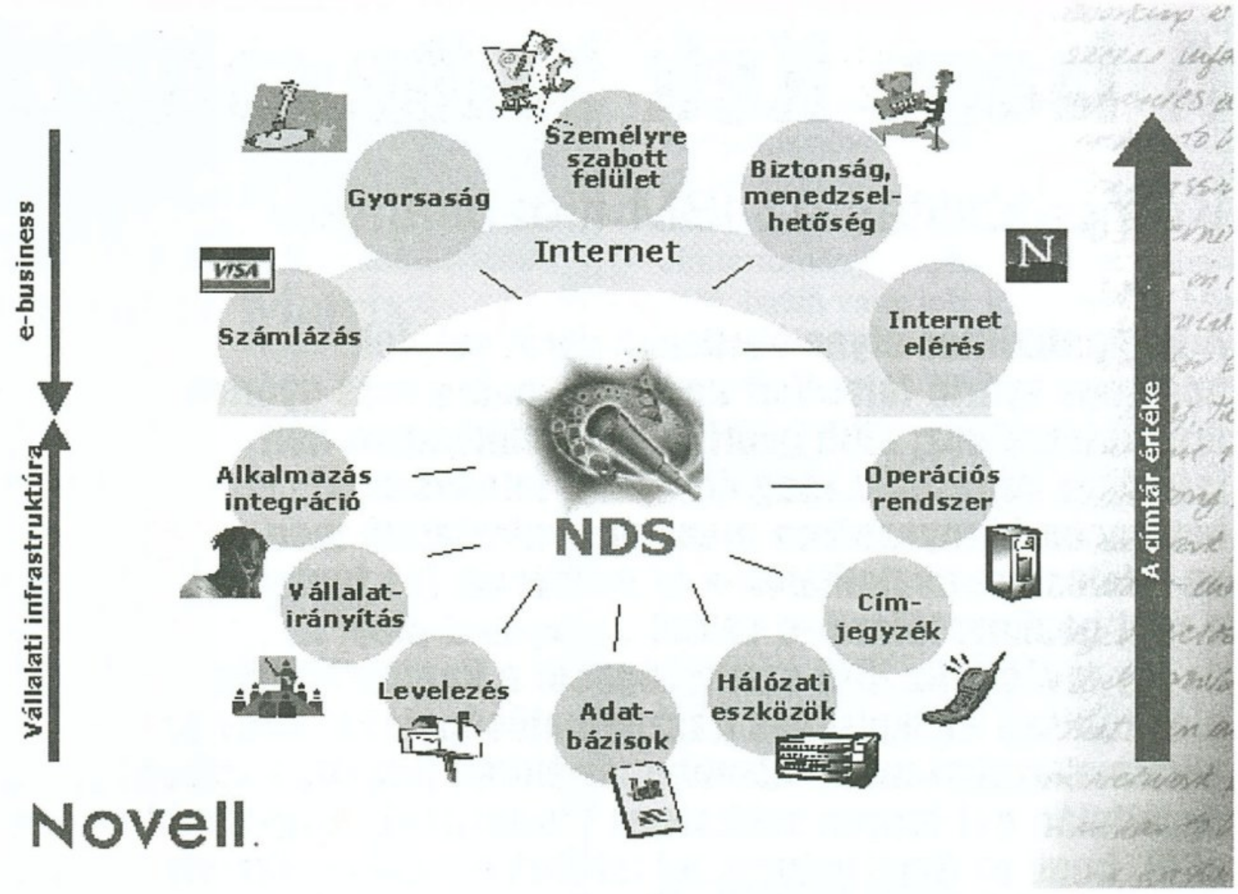
A címtár hatékonysága

Ha szabványos elemekből építkezünk, akkor adataink, szolgáltatásaink gyakorlatilag bárhonnan elérhetők lesznek. Egy egész vállalatot átfogó rendszer kiépítése és üzemeltetése azonban komoly tervezést igényel. A hálózatot úgy kell megtervezni, hogy tükrözze a



felhasználók, felhasználói csoportok igényeit, alaposan végiggondolva, hogy ki mit használ, milyen információt akar elérni. Egyúttal pedig a különböző helyek közötti kapcsolatot átviteli kapacitás és költség szempontjából is optimalizálni kell.

Ahhoz, hogy a vállalat minden alkalmazottja használni tudja a teljes informatikai rendszert, a cég szervezeti felépítése szerint strukturált adatbázist kell kiépíteni, aminek alapján a felhasználók azonosítása és az erőforrásokhoz való hozzáférés jogosultságának menedzselése megtörténhet. Ezt az összetett funkciót a címtárszolgáltatások nyújtják, így az egységes, hatékony vállalati hálózat megvalósításához a vállalati címtár kialakítása az egyik legfontosabb teendő. Mivel az informatikai rendszerek heterogén jellegűek, ezért érthető módon az egyszerre több platformot támogató NDS (Novell Directory Services) vált a legelterjedtebb hálózati címtárszolgáltatássá.



Kiszolgáló platformok

A kompatibilitás elvben azt is jelenti, hogy programjaink a többféle platformról elérhetők, és mi nyugodtan választhatjuk a számunkra legmegfelelőbbet. A valóságban ez persze nem egészen így működik. Mindig újabb platformokat vagyunk kénytelenek a választott alkalmazás miatt bevezetni, és ez többletadminisztrációt jelent, gondoljunk csak például a felhasználói azonosítók, jelszavak és jogosultságok kezelésére. Ezen paraméterek kezelésének leghatékonyabb módja, ha mindent egy helyen tudunk nyilvántartani, felügyelni. Az NDS multiplatformos jellege révén ez szerencsére megoldható.

Az NDS Corporate Edition tartalmazza az NDS Solarisra, Linuxra és Windows NT-re írt változatát, így már nem szükséges külön a solarisos, linuxos és NT-s felhasználók szerverenkénti, illetve tartományonkénti felügyelete. A felhasználók a szoftver telepítésekor automatikusan átkerülnek az NDS-be, a hozzáférési jogok hozzárendelődnek az ott esetleg már meglévő NDS-es felhasználói azonosítóhoz, és ettől kezdve minden hozzáférési jogot egy helyen, a Novell címtárban tudunk kezelni.

Alkalmazások szinkronban

Az alkalmazások kiválasztásánál a megfelelő platform mellett más szempontokat is érdemes figyelembe venni. Ha kialakítottunk egy címtárstruktúrát hálózatunk hatékony felügyelete érdekében, akkor ahhoz illeszkedniük kell az alkalmazásoknak.

Az alkalmazások többféle módon kapcsolódhatnak az NDS-hez. Használhatják azokat a nyilvános API-kat, amelyeket a Novell mindenki számára elérhetővé tett, ezáltal integrálhatók például a Novell, az Oracle, a Computer Associate termékei, illetve a kisebb cégek közül a német Tobit faxszervere, vagy a magyar OfficeNet levelezőrendszere.

Egy másik lehetőség az LDAP protokoll használata, illetve az azon keresztül megvalósuló címtárhozzáférés. Az LDAP könnyített címtárhozzáférési protokoll, a címtárakkal való kommunikáció szabványa. A Directory Interoperability Forum keretében rengeteg szoftverfejlesztő cég (például IBM, Novell, Netscape, Oracle, Lotus stb.) kötelezte el magát a szabvány támogatása mellett. Ez a gyakorlatban azt jelenti, hogy a szervezetet támogató cégek LDAP-kompatibilis szoftvereket írnak, és mivel az NDS natív LDAP címtár, ezért az összes ilyen program az NDS-sel integrálódik. Jó példa erre a Netscape két internetes kiszolgálóprogramja (az Enterprise webszerver és a Collabra News szerver) vagy az IBM WebSphere.

A cégek manapság különböző gyártóktól származó alkalmazások sokaságától függenek. Sok esetben ezen alkalmazások mindegyike saját, különálló címtárat használ, melyet a hálózati szakembereknek kell szinkronban tartaniuk. A Synchronicity-család termékeivel hatékonyabban felügyelhetők a hálózati adatok, hiszen minimálisra csökkentheti a kézi frissítéseket és redundáns munkákat. A program szinkronizálja a hálózat különálló címtárait, ezért a frissítéseket csupán egyszer kell beírni, de a végfelhasználónak is csak egyetlen jelszóra kell emlékeznie ahhoz, hogy bejelentkezzen a hálózatra, az összes operációs rendszerre és alkalmazásra.

A Synchronicity terméksorral megoldható a Notes, az Exchange, az NT Domain és a NetWare Bindery NDS szinkronizációja. Ha az általunk használni kívánt program egyik említett csoportba sem tartozik, akkor a Novell Single SignOn terméke révén tehető NDS-integrálttá.

Ez a termék a már meglévő Novell címtárszolgáltatás (NDS) jelszavával biztosítja a felhasználók számára, hogy csak egyszer kelljen bejelentkezni a hálózatra, és utána hozzáférjenek a különféle platformokon futó alkalmazásokhoz. Minden egyes felhasználó biztonsági profilja — beleértve a bejelentkezési azonosító eszközök és hozzáférési jogok adatait — központilag, az NDS-ben tárolódik. A Novell sok termékhez (például Lotus Notes kliens, terminálemulátor szoftverek) biztosítja a Single SignOn integrációt, egyéb termékek pedig a termékben szintén megtalálható API-k révén saját fejlesztéssel NDS-integrálttá tehetők.

A kompatibilitás a Novell számára olyan hálózati infrastruktúra kiépítését jelenti, amely egyrészt biztosítja a hatékony számítógép-használatot és felügyeletet, másrészt pedig a vállalat igényeinek megfelelően a legkülönfélébb gyártóktól származó hardver- és szoftverelemekkel könnyen bővíthető. Ez az infrastruktúra jelenleg az NDS-sel alakítható ki a legkönnyebben.

Hargitai Zsolt
zsolt_hargitai@novell.com

A Nagy Kék színeváltozása

Kompatibilitás az IBM filozófiájában

A kompatibilitás olyan kérdések sorát veti fel, hogy egy gyártó termékei kapcsolódnak-e más gyártók produktumaihoz; több gyártó együttműködésre ítélt termékei képesek-e megbirkózni a feladattal; heterogén környezetben mennyire valósítható meg az adatcsere; csatolhatók-e új perifériák (és funkciók) a régi géphez; létezik-e valódi „plug-and-play"... és így tovább. Az IBM szemszögéből a kompatibilitás alapkérdése az egységes használhatóság, tehát hogy az alkalmazásokat minél kisebb mértékben (ideális esetben egyáltalán ne) legyen szükséges módosítani, függetlenül attól, hogy az ügyfélgépen, a kiszolgálón vagy valamely köztes eszközön hajtódnak-e azok végre.

Az utóbbi harminc-negyven év tapasztalata azt mutatja, hogy az igazán fontos rendszerekhez senki nem szeret hozzányúlni, ha már sikerült azokat megbízható működésre rábírni. Sőt: nagyon sok helyen ezek az aggastyánkorba lépő alkalmazások jelentik egy vállalat működésének alapját. Mindegyiket meg lehetne ugyan írni újabb, divatosabb nyelven, más környezetre alapozva, mégsem teszik. Azért nem, mert nem éri meg. Az esetleg nyerhető funkcionalitás nincs arányban a szükséges befektetésekkel, és sok minden más úton is hozzáférhető. Nem mellékes az sem, hogy az új rendszereket ugyanúgy hónapokig kell állítgatni, mire annyira megbízhatóan működnek, hogy használatba lehet venni őket.

A konkurencia csak egérkattintásnyira

Nagyon jellemző példa, hogy a német Dresdner Bank egy üzleti integrációs projekt keretében a magyar T-Logic programozóival köttette össze nyugat-európai leányvállalatainak rendszereit, mert ez jobban megérte, mint újraíratni az egészet. Ma a rendszer egységes egésznek látszódik, miközben a háttérben a jó öreg, megbízható rendszerek önállóan teszik a dolgukat. A fentiekből levezethető a hardver- és szoftverváltási ciklusok különbsége is. Az alkalmazások továbbvihetők azt követően is, hogy a kiszolgáló hardver befejezte pályafutását.

Még a legkorszerűbb e-business világ sem teremt olyan helyzetet, hogy bárki megfeledkezhessen a kompatibilitási kérdésekről. Az új rendszerekkel kapcsolatos elvárások nem teljesítése az egész üzletmenetre kihat. A megbízhatóság, a folyamatos készenlét az internetes működés alapja (az internet felett nem nyugszik le a nap). A skálázhatóság vagy az alkalmazások teljesítményének növelése ugyanilyen súllyal esik latba, hiszen ha egy rendszer leáll, a felhasználó számára a konkurencia csak egy egérkattintásnyira van.

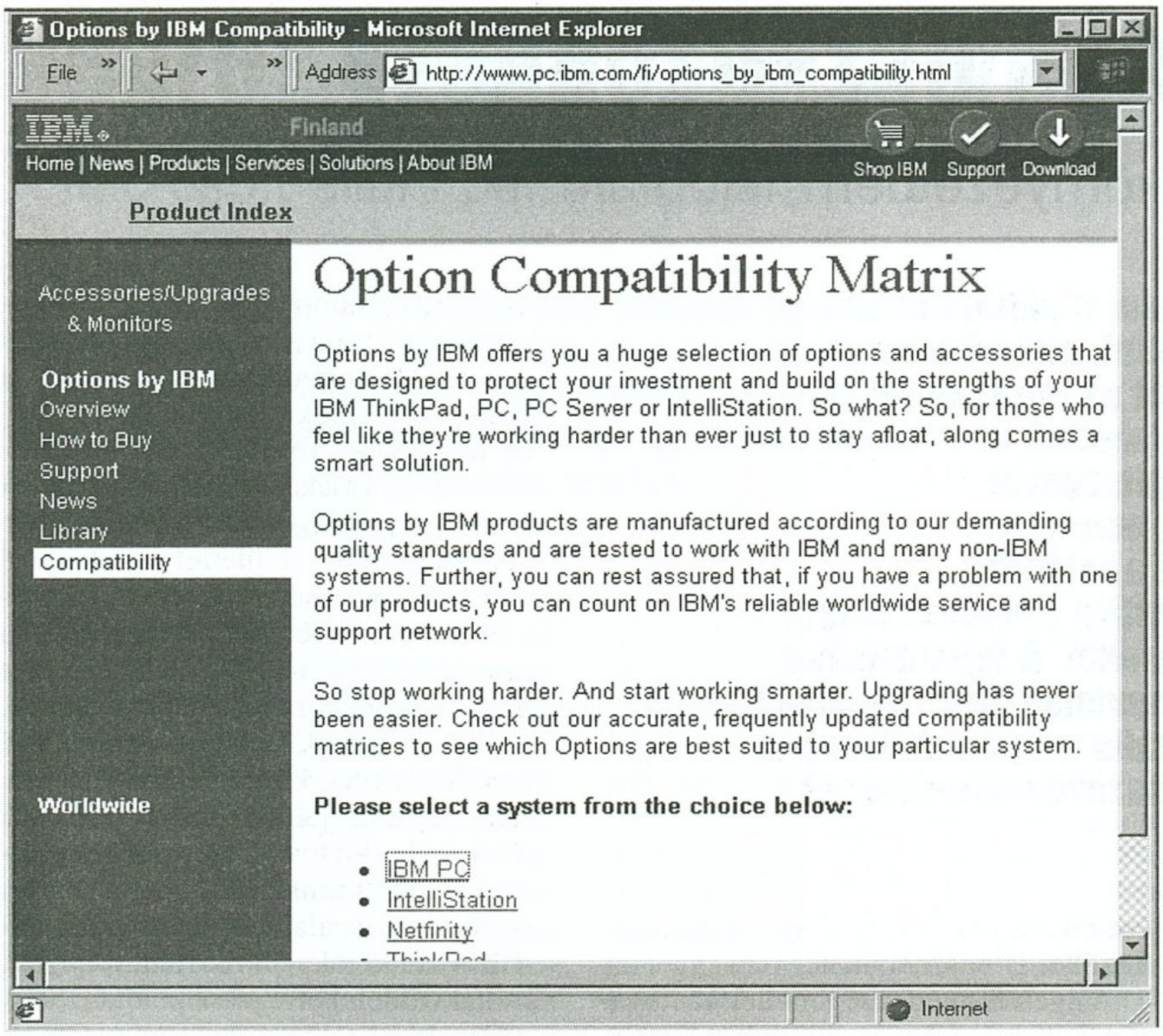
Az IBM azért indul ki az e-business alkalmazási keretrendszerből (e-business Application Framework), mert ez tükrözi jól a mai világ követelményeit, a multiplatformos támogatást, és a hirtelen jelentkező igényekre való reagálást. Ezeknek csak úgy lehet megfelelni, ha betartjuk a legfontosabb összekapcsoló szabályokat — vagy ha úgy tetszik a szabványokat. Ezek pedig a Java, az XML, a TCP/IP. Mindegyiket hardverközelben, de legalábbis az operációs rendszer vagy a köztes szoftver szintjén támogatja az IBM összes saját platformja (OS/2, OS/400, AIX, OS/390 és Dynix). A szoftverek oldaláról a támogatás kiterjed a Linux és a Windows több verziójára, a Sun Solarisra, a HP Unixra, a Novell NetWare-re. Egy konkrét példa: az üzleti integrációban alapvető (és a konkurencia által is elfogadott) IBM MQSeries termékcsalád több mint 35 platformot támogat.

Középpontban a szabványok

A szabványok betartása teszi lehetővé, hogy ha valaki például Linux vagy Windows NT rendszeren kezdi el fejleszteni legújabb termékét, sikeres kivitelezés esetén villámgyorsan átteheti azt egy nagyobb teherbírású platformra, miközben maga az alkalmazás ugyanaz marad. Erre az IBM platformok segítségével minden lehetősége megvan. Már a Netfinity PC-kiszolgálók is kiemelkedő teljesítményt produkálnak a nagygépektől átvett technológiák segítségével, és ahogy nő a terhelés, úgy haladhatunk felfelé a középkategóriás (AS/400, RS/6000, NUMA-Q) kiszolgálókon át egészen az S/390-es mainframe-ekig. Ezek a lehetőségek nagyrészt akkor is megmaradnak, ha beleszerettünk egyik vagy másik IBM platformba, mert a rendszerek extra szolgáltatásait kihasználó szoftvereket sem kell újraírni, ha az adott platformon belül szeretnénk elérni a szükséges teljesítménytöbbletet.



E-business alkalmazási keretrendszer



Annak idején, amikor az AS/400 gépeknél áttértek a 64 bites működésre, az alkalmazások nem is „tudtak" arról, hogy ők most már 64 bites platformon futnak, csak az ebből származó tetemes teljesítménynövekedést lehetett kimutatni. Mivel a memóriacímzéseket az OS/400 automatikusan rendbe rakta, ezután az alkalmazások azonnal teljes értékű, 64 bites alkalmazásokként tudtak működni.

Mikor kell a 64 bit?

Ha már itt tartunk, essék néhány szó a 64 bitességről. Egy rendszer akkor 64 bites, ha minden alkotóeleme az. Tehát nemcsak a felturbózott processzor, hanem az operációs rendszer is, az alkalmazás is... Más kérdés, hogy hol van erre valójában szükség? Nagyon jól hangzik a 64 bites világot propagálni. De arra is választ kell adni, hogy műszakilag mikor szükséges, és mire jó.

Miért nincs például a nagygépeken (mainframe-eken) 64 bites címzés, holott azokat eleve sok felhasználó kiszolgálására tervezték? Azért, mert a horizontális címzést évek óta nagyon jól megoldották, a valós tár ma 24 GB-ig növelhető, és a felhasználók ezreit lehet ezzel a technológiával kiszolgálni. Lesz a nagygépeken 64 bites címzés? Minden bizonnyal, ha majd a teljesítménymérések azt mutatják, hogy erre a továbblépésre szükség van rá.

Önmagában a 64 bit megléte vagy hiánya nem jelenti egyik vagy másik operációs rendszer fölényét vagy lemaradását. Az OS/390 rendszer horizontálisan növelhető, a Unix vagy az NT pedig vertikálisan.

A mai Intel és RISC architektúrák mind virtuális tárkezelést használnak, tehát egy PC-s szerverben van mondjuk 512 MB valós memória, ugyanakkor a Windows NT 4 GB virtuális címtartományt szimulál, és ezen osztozkodik az operációs rendszer és az összes alkalmazás.

A megoldás lényege a merevlemezre kirakott „swap file", illetve a lapozás. 32 bites címzéssel 4 GB tárat lehet címezni, legyen az valós vagy virtuális. Ma a PC/Unix rendszerek kezdik kinőni a 4 GB-os határt, ezért olyan fontos a vertikális növekedés, a 64 bites virtuális tár bevezetése. Az S/390 esetében több címtartomány működik, és 2 GB-os címtartományt kaphat minden felhasználó, batch vagy job. Ez az MVS (Multiple Virtual Storage) operációs rendszernek (ma OS/390) az alapja, mely technológiát az IBM a 70-es évek végén fejlesztette ki. 1990-ben az architektúra tovább bővült, és már nemcsak a programoknak, hanem az adatoknak is külön címtartományokat lehet létrehozni, tehát például egy adatbáziskezelő nagy adatbázis esetén számtalan 2 GB-os címtartományt kezelhet, ami horizontális növekedést tesz lehetővé. Itt tehát nem annyira égető a 64 bites, vertikális növekedés.

A változás örök...

A fentiekből látható, hogy mire akar célozni a cím. Az IBM valaha sok időt töltött el úgy, hogy be volt zárkózva saját világába, világaiba... ennek minden előnyével és hátrányával. Ma meg ott tartunk, hogy az IBM több szoftvert gyárt Windows NT-re, mint a Microsoft maga, mintegy 3000 fős Java programozói gárdát foglalkoztat, többet, mint a Sun... Az IBM emellett sok energiával és költséggel ingyenes szoftvereket fejleszt a linuxos közösségnek.

Az a sok platform, ami az IBM-nél házon belül és az ügyfeleknél megtalálható, a nyílt szabványokon alapuló fejlesztéseket igényli. Az IBM számít arra, hogy a nagy felhasználók elég tudatosak, amikor beruházásaik értékét meg akarják óvni. Kompatibilitás minden szinten — ez az IBM filozófiája. Hiszen csak a változás állandó.

Gacsal József
gacsal@hu.ibm.com



— Ma megint nem vagyok kompatibilis önmagammal...

Integrálás Lotus eszközökkel

Konnektorok a fejlesztőkörnyezetben

Amikor integrált keretbe kell foglalni a vállalatoknál kialakult heterogén adatrendszereket, előtérbe kerül, hogy milyen alkalmazásokkal lehet az információkat vegyes környezetben az üzleti folyamatok szerves részévé tenni. Ilyen a Lotus Domino is. Segítségével olyan alkalmazások fejleszthetők, amelyek az interneten elérik és használják a relációs adatbáziskezelőket, a vállalatirányítási alkalmazásokat és a tranzakciófeldolgozó rendszereket. A fejlesztéshez natív adatbázismeghajtók állnak rendelkezésre, amelyek mind a hagyományos, mind a vizuális programfejlesztési eszköztárral támogatják az alkalmazások elkészítését.

A nyitottság és a sokirányú felkészítettség jegyében alakították ki a Lotus vállalati integrációs (Enterprise Integration, EI) stratégiáját és architechtúráját. A nyitottság kulcsfontosságú az újabb külső rendszerekhez való kapcsolódáskor, a felkészítettség pedig ebben az esetben azt jelenti, hogy megvannak azok az alapok, amelyekre a Lotushoz vállalati szintű alkalmazások építhetők.

A stratégia elemei: egységes objektummodell; program alapú, illetve kód nélküli fejlesztés; nyelvfüggetlenség; eltérő rendszerek támogatása; a vállalati szintű megoldások.

Az egységes objektummodell segítségével a stratégia rendszerezett és összefüggő lesz, miközben csak egy objektummodellt és API-t (Application Program Interface) kell megtanulni, és ez — függetlenül a háttérrendszertől — újra felhasználható. Ezzel a Lotus alkalmas felületet kínál a külső fejlesztőknek a kapcsolattartási lehetőségek bővítésére, és a különböző komponensek több más alkalmazásban is felhasználhatók. A program alapú fejlesztés leegyszerűsíti a gyakori feladatokat, lehetővé teszi a bonyolultabb munkák elvégzését, viszont nincs mindig szükség a kódok manuális megírására, és ilyenkor a kód nélküli (form based) fejlesztés módszerei megfelelőbbek.

Mindkét említett megoldás akkor igazán hatékony, ha az alkalmazások fejlesztése nem kötődik egyetlen programozási nyelvhez. A programozási felületet a nyelvfüggetlenség szem előtt tartásával célszerű egységesen megtervezni, ami nem zárja ki, hogy szükség esetén a fejlesztők nyelvspecifikus eszközöket is használjanak.

A nyelvfüggetlenség mellett támogatást kell biztosítani a különböző rendszereknek. A már említett relációs adatbáziskezelők mellett ide tartoznak az ERP (Enterprise Resource Planning) szoftverek, címtárak, tranzakciós rendszerek. A nyílt felület lehetővé teszi a későbbi bővítéseket, alkalmazkodni lehet a vállalatok igényeihez méretezhetőségben, helyi beállíthatóságban.

A Lotus EI architektúrához tartozó kapcsolók (konnektorok) más rendszerekhez nyújtanak egységes kapcsolódási felületet. Az eszközök (tools) a magasabb szintű funkciók elérését teszik lehetővé.

A Lotus általános kapcsolója, az LC (Lotus Connector) több programozási nyelven keresztül is elérhető (LotusScript, Java), API-ja alkalmas a rendszerek közötti adatmozgatás elvégzésére. Az LC API Toolkit révén külső fejlesztésű konnektorok is kialakíthatók. Az EI eszközök pedig ezeknek a kapcsolóknak a felhasználásával működnek. A fejlesztéskor egységes, űrlap jellegű kezelőfelülettel támogatják a kódírás nélküli alkalmazásfejlesztést, és olyan vállalati funkciókat is kínálnak, mint ütemezés, naplózás, távoli felügyelet.

Az EI eszközök konnektorai egységes objektummodellt kínálnak, és az ezen alapuló alkalmazások magasabb szintű változtatás nélkül is megteremtik a különböző rendszerek közötti kapcsolatot, ha a külső metaadat vagy az adatelérés jellege nem különbözik. Ugyanakkor minden Lotus kapcsoló és eszköz támogatja a multinacionális környezetet, automatikusan felismeri a nemzetközi karakterkészleteket és képes ezek között fordítani. Ennek ellenőrzésére a termékeket nemzetközi környezetben is tesztelik.

Az egységes felülettel rendelkező natív Lotus Connectorok közül az általános eszközök ODBC és SQL alapú kapcsolattartást tesznek lehetővé, és vannak konnektorok a DB2, az Oracle, az SAP, a Tuxedo rendszerekhez is. E konnektorokkal kihasználhatók a termékspecifikus jellemzők. Az LC API architektúra biztosítja a csatlakozók által támogatott funkcionalitás belső azonosítását, és az alapkészleten kívül specifikus funkciókat is be lehet építeni. Ezáltal válik lehetségessé például, hogy az Oracle csatolór támogatja az Oracle kötegelt átviteli képességét, és a Notes konnektor használható a csatolt állományokon (file attachments).

A Lotus Enterprise Integrator 3.0 (LEI) a Lotus NotesPump 2.5a utóda tartalmaz különféle adatbázis, címtár és állományrendszer konnektorokat, valamint metakonnektorokat. A Notes alapú kezelőfelület lehetővé teszi az űrlapra épülő kapcsolatok létrehozását, amelyek a LEI Server számára tartalmaznak utasításokat. A Domino Enterprise Connection Services (DECS) a Domino 4.63 óta minden új verzióban megtalálható, és általa a külső adatok mélyen, valós időben integrálhatók a Domino alkalmazásokba. Az alacsony szintű Domino szervereseményeket „elkapva" a DECS adatbehívó vagy módosító utasítást ad ki a konnektorokon keresztül. A többszálú, párhuzamos kapcsolatok a külső rendszerekkel igen nagy teljesítmény elérését teszik lehetővé. (További információk olvashatók erről a http://www.lotus.com webhelyen.)

A Lotus EI stratégiája és architektúrája megoldást nyújt a Lotus külső rendszerekkel történő összekapcsolásakor, mert a konnektor hatása nem korlátozódik a LotusScriptre, és segítik az ún. metakonnektorok is, melyek mintegy „becsomagolják" a kapcsolót, és elvégzik az adatátalakításokat a csatlakozó felületeken.

Zsigmond Olivér
ozsigmon@dominomail.com

Végre! Végre! De mivégre?

A Microsoft elleni per margójára

A Microsoft bűnös a Sherman-féle trösztellenes törvény megsértésében — hirdette ki Thomas P. Jackson bíró a verdiktet. Ezt ki-ki saját meggyőződése szerint történelmi pillanatként vagy iparági gyásznapként értékelte áprilisban. Szokásom szerint én kissé másként látom...

A Microsoft legfőbb tulajdonosa és szellemi atyja természetesen az iparágat tönkretevő lépésként próbálta interpretálni az elmarasztaló bírói döntést, és arra kérte a részvényeseket, lobbizzanak a cég érdekében. Tette ezt annak ellenére, hogy valószínűleg ő is tudja: ez az ítélet a világ működése, de tulajdonképpen még a cég szempontjából is súlytalan. Arról meg semmiféle döntés nincs — és ismervén a jogászokat, talán évekig nem is lesz —, hogy milyen következményei legyenek a cég életére. Feldarabolják? Kivonják a forgalomból néhány termékét? Kötelezik bizonyos kódok közzétételére? Ki tudhatja ezt előre! Ráadásul bármi lesz is a „büntetés", azt az egyetlen döntést, ami valóban jelentene valami pozitívumot az informatika számára, nem lehet meghozni, és nincs is olyan erő, amely a betartását szavatolhatná vagy akár csak ellenőrizhetné.

Mire gondoltam? Az jelentene valamit a Microsoft manipulációinak áldozatul esett dolgok újraélesztésében (ezek: programozói precizitás, gondos fejlesztés, átgondolt verzióváltás, a felhasználók valós igényeinek megfelelő programok készítése, a szabványok szigorú betartása stb.), ha ki lehetne kényszeríteni: a programok ne a marketing, hanem az informatika szabályai szerint készüljenek, és csak tényleg kész állapotban kerüljenek piacra, akkor is valódi értékükön.

Mindannyian tudjuk, mit értünk e fogalmakon, de gyanítom, hogy senki nem tudna egységes, mindenki számára megfelelő szabálygyűjteményt felállítani, melynek mentén ezt be lehetne tartatni, mert félő: be is fagyasztaná a fejlődést. Nem marad más, mint mindenkinek a lelkiismeretére és szakmai önérzetére bízni, hogy megtalálja a határt a még elfogadható és a már „bűnös" módszerek között. Aki meg a többiek egybehangzó véleménye szerint gyakran és látványosan átlépi ezt a láthatatlan határt, azt meg kell fékezni.

Persze valahol érthető Mr. Gates véleménye, hisz az egyébként „józan" jelzővel illetett ún. „Pénzügyi Világ" is elég hisztérikusan reagált az elhangzott ítéletre. Azonnal esni kezdett a cég részvényeinek ára (Mr. Gates elkezdett visszacsúszni a második helyre Amerika leggazdagabb üzletembereinek sorában), sőt eleinte más „technológiai papírok" értéke is megindult lefelé. Pedig a Microsoft által jegyzett programok egy bittel sem lettek rosszabbak ezen a napon (igaz, jobbak sem), mint azelőtt voltak.

Mellesleg: a „technológiai ipar" meghatározás úgy marhaság, ahogy van, hisz minden iparnak van technológiája, nem is egy, de a technológiának ipara!? A technológia vagy technika fogalmát a számítástechnikára, vagy akár az elektronikára leszűkíteni szellemi vakság.

Hogy a Microsoftot végre bűnösnek nyílvánították a szabadpiac alapszabályainak megsértésében, úgy értékelni, miszerint az egész „iparág" megindult lefelé a lejtőn, nem éppen a józan gondolkodás megnyilvánulása, inkább „technológiai" hisztériaként tudom elkönyvelni magamban. Arra jól rámutat a jelenség, hogy az informatikai cégek egy részének gazdasági jelentőségét alaposan túlértékelték, illetve arra is, hogy azok a mesés vagyonok (úgy is mint az irigység tárgyai) valójában léggömbök, nem valódiak. A papírok mindig annyit érnek, amennyit a világ adott pillanatban hajlandó lenne adni értük. Ha egy ilyen kis tűszúrás, mint a cikk elején hivatkozott ítélet, ekkora „földindulást" képes előidézni, akkor ez az érték virtuális. Valódi gazdasági hatalom helyett csak „marketing", reklám, légvár...

Lehet valakinek a birtokában sok millió részvény, abból nem tud enni, sem enni adni a világ olyan részein, ahol az ilyen „pénznek" nincs értéke: háborútól, éhezéstől, árvíztől, aszálytól, járványtól sújtott országokban. A világ valódi tragédiái ott történnek. Az, hogy most a Microsoft esetleg kénytelen lesz beismerni: nem igazán becsületes eszközökkel pumpálta oly nagyra a „birodalmat", nem old meg semmit a Föld nagy problémái közül. Ettől a háborúkból sem lesz kevesebb, a légkör sem lesz kevésbé szennyezett, sőt a dodómadár sem kel életre többé. Még az sem jelentene olyan nagyon sokat, ha az egész céget felszámolnák. Egy csomó embernek egzisztenciális katasztrófát okozna (talán még nekem is), hisz világszerte sok tízezren abból élnek, hogy megpróbálják eladni a cég termékeit, más tízezrek pedig abból, hogy megpróbálják a rájuk sózott, néha ipari hulladéknak is csak nagy jóindulattal minősíthető rendszereket nap mint nap életben tartani, a cég nagyobb dicsőségére. A világ többi része vidáman haladna tovább.

Túlzott önbizalomra vallana, ha magunkat, a mi kis szűk világunkat (a számítástechnikusokét, az informatikusokét) azonos súllyal értékelnénk, mint amekkora problémahalmazt a föld és a rajta élők sorsa jelent. Valóban nagy hatása van ennek a rétegnek a világra, de nem azonos vele, csak egy igen kis részhalmaza annak. Megvan a szerepe az élet menetében, nélküle más (talán rosszabb, talán jobb) lenne minden, de vegyük észre: az emberiség megvolt már az elektromosság felfedezése előtt, és jó része ma is megvan elektromos alapszolgáltatások nélkül is!

Nagyon sok olyan ember él, aki soha nem hallott a Microsoft peréről, de még magáról a cégről, sőt esetleg a számítógépről sem. Vagy ha hallott is, legfeljebb annyira érdekli az egész, mint engem mondjuk a csíbor szaporodásbiológiája. Még az is megeshet, hogy e „tudatlan" emberek egy része közben naponta használ számítógépet, számítástechnikai eszközt, és nem is tudja, mi az. Ma már „mikrokompjúterek" dolgoznak nemcsak a mobiltelefonokban és a telefonközpontokban, hanem a komolyabb mosógépektől kezdve megannyi elektromos és elektronikus eszközben. Autóinkban, tévéinkben, napi használati tárgyainkban.

Ez az iparág már messze túlmutat a PC-ken, és utólag legfeljebb mint érdekes epizódra emlékszünk majd vissza, hogy egy cég a játékszabályokat megszegve igyekezett magának sokkal nagyobb tortaszeletet kihasítani, mint amekkora tényleges teljesítménye alapján megilletné őt, és ezt az amerikai társadalom egy része végül nem volt hajlandó tolerálni. Rövidlátás vagy önsajnálat az említett ítélet súlyát azonosítani ezzel a mérhetetlenül nagy jelentőségű iparággal, és annak „végnapjait" jósolni csak azért, mert egy pimasz, a törvényeket, a szabályokat és a szabványokat felrúgó cégnek egy kicsit az orrára koppintanak, hogy eddig, és ne tovább.

Ha valakinek az a célja, hogy cégét megkerülhetetlenné teszi a számítógépes ipar minden ágában, és ezt úgy éri el, hogy brilliáns programokkal árasztja el a világot, ha jól használható szabványokat dolgoz ki, ha olcsó nyílt rendszerekkel könnyíti meg a többiek dolgát, akkor ennek mindenki örülne. Halála után talán még szentté is avattatnák.

Ha valaki ehelyett arra a „zseniális" felismerésre jut, hogy a programok megírása nehéz, sokba kerül, és a precizitással, a megbízhatósággal nem lehet olyan jól keresni, mint az ügyesen reklámozott, de hevenyészve összecsapott programokkal, azt a többiek bizonyos idő után nem fogják eltűrni. Egy cég termékeinek nélkülözhetetlensége illúzió. Amíg a piac tényleg szabad, addig a jó és új dolgok előtt is nyitva áll az út. Az internet is megszületett, és sokáig jól megvolt a Microsoft nélkül, sőt annak törekvései ellenére fejlődött, amire mostanában nem mindenki szeret emlékezni. Mr. Gates valami olyasmit mondott néhány éve, hogy ha lenne egy gomb, ami megszüntetné az internetet, ő gondolkodás nélkül megnyomná...

A reklám és termékminőség dilemmáján túl más történelmi hagyományokból is merített Mr. Gates: sok pénzzel mindenféle disznóságot el lehet fedni. Arra is rájött, hogy a kis törvénytelenségek nagy technológiai áttörésként tüntethetők fel, nem kell hozzá más, mint sok jó propagandista, nagyszabású reklám és a cég „küldetéstudatának" hangsúlyozása. Arra is rájött, hogy az emberek egyszerre kapzsik és lusták (ami nem is olyan nagy felfedezés), és ebből pénzt lehet csinálni. Odaadni nekik valamit ingyen, azt megszokják, használják, később pedig már aránytalanul nagy összegért is megveszik, mert valami újabbat tanulni és arra átállni elég nehéz. Időbe telik és fárasztó. Nagyobb aktivitást követel, mint a tévé előtt ülni és bambulni, vagy az egeret kattintgatni.

Nem véletlen, hogy alig egy évtizede a — mellesleg a BSA-t életre hívó — Microsoft még fennen hirdette: nem akar programjainak alkalmazásáról a „kisemberekkel" vitatkozni. Ha valaki használja azokat, örülnek neki, és kész. A drogkereskedő is hasonlóan dolgozik: rászoktat az árura ingyen, majd amikor már nem tudsz szabadulni, lecsap rád, pénzt kér, később követel, még később kényszerít. A drogdíler őszintébb: nem hivatkozik törvényekre, amelyeket ő hozatott meg a maga felé hajlított politikusokkal, és nem is rendőrt hoz rád. Egyszerűen és nyíltan megver, megveret. A végeredmény azonos: a beetetett áldozat innen kezdve akkor is tejel, ha az esze már tiltakozik, csak ösztönei és megszokásai kívánják a „szert". Ezekből a tanulságokból született meg a csodálatos módszer:

— Keress egy jól bővülő piacot, és terítsd be olyan termékekkel, amelyek úgy-ahogy működnek. Tedd őket külsőségeikben vonzóvá, és költs sokat a reklámjukra.

— A termékhibák kijavítása helyett mindig hozz ki újra eladható „jobb" verziókat, ezekbe építs be újabb hibákat, melyeket a későbbi verziók fognak kijavítani.

— Ügyelj rá, hogy a régi és az új verziók ne tudjanak egymással szóba állni.

— Gondoskodj róla, hogy a régi verzió az új megjelenése után törvényesen már ne legyen megszerezhető. A törvényellenes használókra pedig csapj le (persze ne személyesen, hanem helyi rendőröket és politikusokat alkalmazva).

— Vedd rá a kormányokat — zsarolással vagy „ajándékokkal" —, hogy tegyék szabvánnyá a programjaid által generált dokumentumokat az államigazgatásban. Gyakori formátumváltozással pedig igyekezz elérni, hogy csak a te programjaid tudják olvasni azokat.

— Ha valamelyik (leendő) konkurens jó dolgot talál ki, vedd meg az egész céget, számold fel, és termékét add el sajátodként.

— Mindig módosítsd egy kicsit a szabványokat: csak annyira, hogy ne legyen feltűnő, de legalább annyira, hogy akik az eredetit betartják, ne tudjanak a megváltoztatottal kooperálni. És az egészet nevezd újításnak vagy technológiai áttörésnek.

— Ha olyan nyílt rendszerbe akadsz bele, amelyet nem lehet sem megvenni, sem zárttá tenni, írd át annyira, hogy a kedves mamája se ismerjen rá, majd hirdesd, hogy a tied a szabályos, és az eredeti a hamis.

— Amikor végre nálad is sikerül valahogy kiizzadni (esetleg „megvenni") a másoknál évek óta alkalmazott, de nem elég közismert megoldásokat, köríts azt hatalmas csinnadrattával, és jelentsd ki: mi vagyunk az élharcosok.

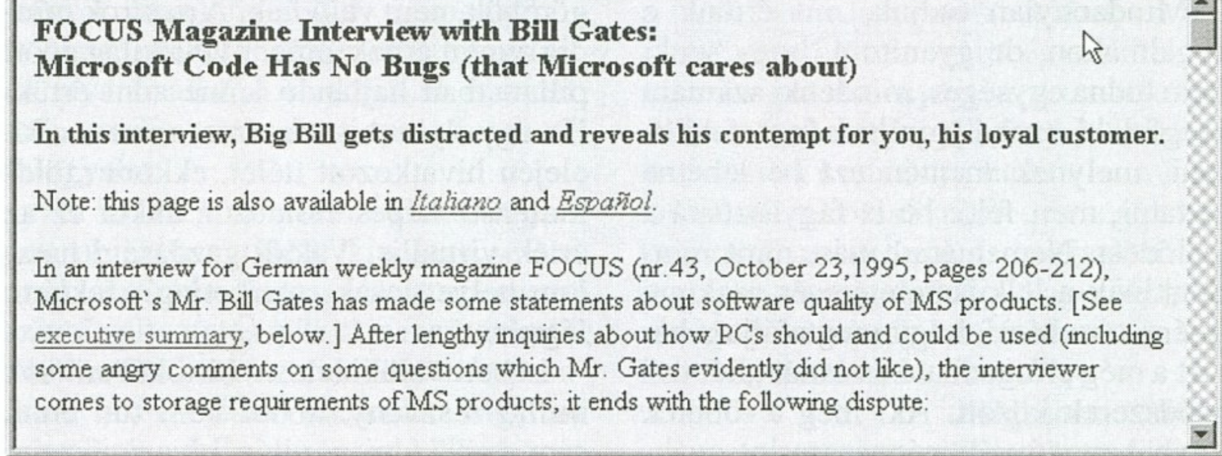
— Ha mindezekért valaki beperel, ne sajnáld a pénzt arra, hogy lemosd a pályáról az ellenfelet. A veszteséget később úgyis megtérítik a vevők, akik elhiszik: az ellenfél nem azért lépett vissza, mert megvetted őt, hanem mert neked van igazad.

— Ha a fentieket minden elterelő manőver ellenére feketén-fehéren rád bizonyítják, kezdj el sírni, és hirdesd a világvégét. Elég sokan lesznek, akik elhiszik... Hát most itt tartunk.

Pedig közben kizöldült a táj, megjöttek a fecskék és a gólyák, tüsszögnek a pollenallergiások. Itt a tavasz, jön a nyár, elvonult az árvíz, a Föld szépen forog tovább, rajta az élet — olyan, amilyen — zajlik a maga módján. Hol minden szép, hol komor, hol békés a táj, hol háború füstje gomolyog. Új élet születik és búcsúztatnak mást, érettségizők izgulnak, szerelmesek andalognak a parkban...

Talán csak Mr. Gates feje fáj sűrűbben, mint szokott. És még mindig nem érti, hogy miért.

Sándor Gábor


http://www.cantrip.org/nobugs.html - Microsoft Internet Explorer

File » Address http://www.cantrip.org/nobugs.html#exec

FOCUS Magazine Interview with Bill Gates:
Microsoft Code Has No Bugs (that Microsoft cares about)

In this interview, Big Bill gets distracted and reveals his contempt for you, his loyal customer.

Note: this page is also available in *Italiano* and *Español*.

In an interview for German weekly magazine FOCUS (nr.43, October 23,1995, pages 206-212), Microsoft`s Mr. Bill Gates has made some statements about software quality of MS products. [See executive summary, below.] After lengthy inquiries about how PCs should and could be used (including some angry comments on some questions which Mr. Gates evidently did not like), the interviewer comes to storage requirements of MS products; it ends with the following dispute:

KERESKEDELMI KIÁLLÍTÁS ÉS KONFERENCIA

Cards Eastern Europe 2000

Information Security iSEC EASTERN EUROPE 2000

A Cards Eastern Europe 2000 a régió egyik legrangosabb kereskedelmi bemutatójává nőtte ki magát, ahol különböző szervezetek bemutathatják termékeiket, szolgáltatásaikat és építhetik kereskedelmi kapcsolataikat Kelet-Európa más hasonló szervezeteivel. Az esemény célja az egyre gyakrabban használt chip és intelligens kártya technologiák bemutatása - amik megváltoztatják a mindennapi életet és az üzletkötések módját.

iSEC Eastern Europe 2000 - Vállalkozásának biztosítása. Az iSEC Eastern Europe' Kiállítás bemutatja a legmodernebb

IT Védelmi megoldásokat és technologiákat, többek között:
Anti-VirusBiometrics, Electronic Commerce, Firewalls, PKI, Software, Secure electronic mail, Secure Operating Systems, Security Audit, Encryption, Key Generation, Systems Verification, Digital Signatures, Smart Cards, Intrusion Detection Systems, Virtual Private Networks, Remote Access, Cryptography

Vállalatának jövője a maga kezében van!

2000 június 20-22

Budapest Kongresszusi Központ · Budapest · Magyarország

A kiállítás nyitvatartási ideje

Kedd, 2000. június 20	**10.00de - 6.00du**
Szerda, 2000. június 21	**10.00de - 6.00du**
Csütörtök, 2000. június 22	**10.00de - 4.00du**

Global Gold Sponsor: HYPERCOM

Event Gold Sponsor: Eurocard MasterCard, Maestro

Gold Sponsor: Computer Associates — Software superior by design.

Global Sponsors: Datacard Group, ORGA — The Smart Card Integrator

Silver Sponsors: network associates — Who's watching your network, PGP

Bronze Sponsors: ISS, Schlumberger, Gemplus

Endorsed by: Hungarian Smart Card Forum

Szervező: AiC Worldwide — Where Business Does Business

Kérjük, töltse ki és küldje vissza: Nikolai Birukov +44 (0)20 7242 1508. **Email** nikolaib@aicworldwide.com

❑ Kérjlük küldjön további információt a támogatás lehetöségeiröl
❑ Kérjük küldje el a teljes Cards Konferencia programját
❑ Kérjük küldje el a teljes iSEC Konferencia programját
❑ Kérjük, küldjenek ingyenes belépöjegyeket a Cards/iSEC kiállításra

Név: .... Foglalkozás: ....

Cim: ....

Telefon: .... Fax ....

Email: ....

Jelentkezzen most: www.cards-worldwide.com www.isec-worldwide.com

AiC Worldwide Tel: +44 (0)20 7242 1548

Üzlet az üzlet a neten

Akinek nem (háló)inge…

Kinn e-business, benn e-business, kell-e mindjárt benn e-bízni(sssz…)? Az alcímben nem azokra célzok, akik ruhátlanul szeretik aludni jól megérdemelt álmukat. Az internethez kapcsolódó néhány érdekes, de egyelőre csak Észak-Amerikában elérhető szolgáltatás és termék közül mazsolázok. Egy részük még kiforratlan, végső tesztfázisban van, de mindenképpen figyelemre méltó a mögöttük meghúzódó ötlet. Mivel a globális világgazdaság részei vagyunk, előbb vagy utóbb bennünket is utolér az e-business végzet, nem árt tehát előre felkészülni… és nem kell magunkat e-bízni(c).

Háló alapú integrált irodai csomag

Miért jó nekünk az Online Office Suite? Az internetről betöltendő szövegszerkesztő, táblázatkezelő, prezentációs szoftver. Kimenetük HTML, azaz a webről megszokott formátum. Egyből webszerverre menthetünk, megjelenítve a weblapon. A PC-s világban a Microsoft Office alkalmazások mindent beborítottak. Dominanciájuk majdnem teljessé vált. Ezért ezek a gyakran internetes portálként bejelentkező szolgáltatáscsomagok adatállomány import/export szintjén kényszerűen támogatják a Microsoft Word, Excel és PowerPoint formátumait. (A nagy reményekre hivatott StarOffice — emlékeznek még? — most éppen a Napon dekkol, a JAVA még hátravan jeligével. Magyarításának egyelőre befellegzett.)

Három kisebb, kevésbé ismert cég megoldására hívnám fel olvasóim figyelmét. Van-e keresnivalójuk ezeknek a cégeknek a végeláthatatlan neten? Addig, amíg a nagyok (Microsoft, Corel, Lotus…) nem alakítják ki nyerő stratégiájukat, mindenképpen igen.

Megjegyzem, hogy lassan már dereng, mi lehet a Microsoft kereskedelempolitikai szándéka. Új licencpolitikája szerint a dobozos termékhez képest a plusz felhasználói engedélyt 10-15 százalékkal olcsóbban kínálja. Tehát igyekszik átterelni a termék dobozos forgalmazását doboztalanra. Ezzel jelentős költség takarítható meg, nem kell legyártatni a karton csomagolóanyagot és annak feliratait, nem kell képet terveztetni, CD-t sokszorosítani, a dobozokat (eladottan, eladatlanul) szállítani, raktározni. Rengeteg anyag és munkabér megtakarítható.

Az előre kifizetett, internetről letölthető terméknek az a nagy előnye is megvan, hogy egyszerű a szoftverkövetés, ugyanonnan férhet hozzá a vevő a frissített változathoz, lehet tudni, hogy honnan kezdeményezték a szoftver downloadot. A gyártó közvetlen kapcsolatba kerülhet a végfelhasználókkal. Ami eddig szükséges rossz volt, azt kiiktathatják.

Normális közgazdasági esetben megszokhattuk, hogy a gyártás, a fejlesztés, a reklám költségeinek nagy részét a nagykereskedelem finanszírozza a gyártó számára, aki így „olcsó” pénzhez jut egy bankhitelhez viszonyítva. Most azonban az értékesítési lánc lerövidül, maradnak az internetes pénzelfogadóhelyek, de lám, azok is bankokhoz kötődnek, így a bankok feje nem fáj. Vajon a kisebb cégek állják-e a sarat és a növekvő fejlesztési és reklámköltségeket a nagy, tőkeerős gyártókkal szemben? Az biztos, hogy a közgazdasági tankönyvek utolsó fejezeteit rövidesen át kell fogalmazni. Globálisan.

Az szoftverproduktum fizikai átalakulását illetően a többiek be fognak állni a nagyok szélárnyékába, és hasonlóképpen cselekszenek, mert nem tehetnek mást. A szoftver egyébként is a leginkább környezetbarát termék. Mindegy, hogy hány megabájt szemétről van szó, egyetlen mozdulattal letörölhető, felülírható. Ártalmakról csak akkor beszélhetünk, ha nem azt töröljük, amit kellett volna…

A MyFreeDesk („szabad íróasztalkám”) fedőnevű Applix Anywhere, a Nuomedia által felkarolt, a Lotustól elszármazott eSuite, valamint az egyetlen abszolút saját fejlesztésű, új ThinkFree mind nyilvános béta fázisnál tart, azaz részeiben vagy egészében kipróbálható, megtekinthető. Mi a véleményem róluk? Nos, amikor hagyományos levelet írunk, abba is belerajzolhatunk valamit, előszedhetünk színes ceruzákat, beleragaszthatunk figurás matricákat, használhatunk mintás, sőt szagosított (ennél az érzékszervi ingernél még lemaradásban van a web!) levélpapírt és borítékot. Mégis nagyon ritkán élünk ezekkel a lehetőségekkel. A színes filcek, képes matricák, ollók és ragasztók ott lehetnek a polcunkon, de ha egyszerű, közönséges levelet akarunk írni, elő sem vesszük azokat. Macerás viszont a



normál levélről másolatot készíteni. Nem írunk indigóval... Az elektronikus levelezés nagy előnye annak kényelmes archiválása, rendszerezett tárolása, gyors visszakereshetősége, papírra történő reprodukálhatósága.

Nyilvánvaló, hogy a két dolgot érdemes ötvözni. Ki-ki saját vérmérséklete és számítástechnikai felszereltsége alapján döntse el, milyen szolgáltatásokkal kíván élni (ma még túl azon, hogy az internetezés sem tekinthető általánosnak, és egy színes nyomtató sem az.)

Piszkos anyagiak. Ma már egy Microsoft Office ára a szárnyaló dollárárfolyam miatt bruttó 150 000 forint fölé nőtte magát, amennyiért egy közepesnél jobb gépet is vehet magának az ember. Csendben jegyzem meg, hogy a Microsoft Worksért még mindig nem kell húszezernél többet kiadni, és ugyanolyan írógép, mint a WinWord. (De ez köztünk maradjon.) A webes online csomagok — tételezzünk fel tárcsázásos internetelérést és nem ingyenes telefont — általában az alapszoftverek letöltésével kezdődnek. Lehet, hogy önmagukban ingyenesek lesznek (maximum valamilyen reklámba botlunk minden harmadik mondat leírása után), de az adatok mentése, mozgatása online csatlakozást fog igényelni. Ez alól kivétel a ThinkFree, amely offline is dolgozik. Tehát marad plusz költségként az internetelérés mindenkori költsége. Ez a közeljövőben minimálisra csökken. Az is netezik, akinek gépén Microsoft Office van (legálisan, azaz pénzért vette), így az online suite felhasználók egy MS Office vagy Works árával mindenképpen beljebb vannak anyagi szempontból. Nem beszélve arról, hogy lemezkapacitásukat sem a gigantikus méretű Microsoft programok pocsékolják el. De ez utóbbi ma még kevésbé fontos.

Csudálatos nyomógomb

Ez a kulcstartónyi kütyü nem más, mint a Xenote cég iTag nevű új terméke, melyet gyerekkori „stipi-stopi" játékunkhoz tudnék leginkább hasonlítani. Ehhez azonban nem kell társ. Ha valami megtetszik, csak megnyomunk egy gombot, és az — pénztárcánktól függően — hamarosan a miénk lesz. Ezzel azt hiszem teljesen összezavartam a kedves olvasót, fogalma sincs, miről van szó. De valljuk be, nem hangzik rosszul ez a csudálatos nyomógomb.

Kicsit előreszaladtam időben és fantáziában, mert egyelőre az iTag csak hangra érzékeny. Megyünk az autónkkal, váltogatjuk a rádiócsatornákat, hogy emészthető zenét találjunk. Hirtelen rálelünk régi kedvencünkre, amelynek szerzőjét, címét, előadóját már elfelejtettük. Gombnyomás az iTag-en, és az szépen rögzíti az éppen hallható zeneszámot. Otthon, számítógépközelben azután elővesszük az iTag-et, rácsatlakoztatjuk a számítógépre, a neten felkeressük a Xenote honlapját, és a felvett hangmintákat összevetjük az ott találhatókkal. Rövidesen megkapjuk a lekapott (lestoppolt) számok pontos adatait, valamint természetesen az adott szerző és előadó megjelent albumainak, CD-inek ismertetését, pénzért megvásárolható relikviáinak kínálatát. Belehallgathatunk, kedvünkre csemegézhetünk. Uccu, azonnal meg is rendelhetjük a kérdéses korongot, vagy a ma terjedőfélben lévő, a fiatalok körében divatos MP3 formátumban is „kivásárolhatjuk", és letölthetjük MP3 lejátszónkhoz.

Hogyan lehet analóg felvételt összehasonlítani digitalizált hangmintákkal? A Xenote egyszerűbb, a számítógép által rutinmunkának tekinthető megoldást választott. Az iTag nem tesz mást, mint gombnyomásunkra elmenti a pontos időt és a rádióállomás vételi frekvenciáját. Igaz, Amerikában több mint ezerkétszáz rádióadó játszási listájával kell számolni, de a központi szerver ezek közül is könnyedén kikeresi a gombnyomáskor játszott nóta adatait. Magyarországon ez sokkal egyszerűbb lenne a kevés rádióadó miatt is. Bár ismerve a hazai dokumentátorok, számítógépkezelők precizitását, biztosan nagy balfogásoknak is tanúi lehetnénk. A gép adminisztrációs hiba (vagy a rendszerdátum elállítása) miatt Lagzi Lajcsit összetévesztené mondjuk a száz tagú cigányzenekarral, esetleg Beethoven sem nyugodhatna a sírjában...

A Xenote fejlesztői nem állnának meg itt. Beépítenének egy vonalkódolvasót a kis kézi megjegyző masinába, és innen kezdve ránk szakadhat a világ. Látunk valami jót, valami szépet, megkeressük a vonalkódját, és katt. Szaladunk egy hipermarketbe mint a fogyasztói társadalom jól nevelt polgárai, és az ott kinyomtatott lista alapján megszerezzük magunknak a kívánt javakat(t). Katt. Még szerencse, hogy egyelőre nem vonalkóddal jövünk a világra, mert akkor szó szerint bekattanhatna ránk bárki, például a metrón... Ellenőrök kíméljenek!

Fáradhatatlan könyvelő

A Microsoft egyelőre vesztésre áll az USA igazságügyminisztériuma és további 29 USA tagállam által ellene indított perben. A tőzsdén a fellegekben járó Microsoft részvények nagyot zuhantak. De nem ezt az ügyet kívánom kommentálni (nem mintha le akarnám becsülni a jelentőségét, de hol van még a vége...), csupán megemlítem, hogy ennek következtében a legvagyonosabb ember stafétabotját akár hosszabb távon is az Oracle cég elnök-vezérigazgatója, Larry Ellison cipelheti. Nos, éppen ő fektet be most milliókat (dollárban) a NetLedger nevű online könyvelési vállalkozásba. Az interneten keresztül könyveltetni... ugye nem hangzik rosszul, és mondanom sem kell, hogy a szoftverrendszer Oracle adatbázison alapul.

Képzeljük el, hogy könyvelőnk a hét összes napján, nullától 24 óráig rendelkezésünkre áll. És nem órabért kap, nem is a maffia tartja sakkban. Havi 5 dolcsiért dolgozik nekünk az interneten keresztül. Akárhol járhatunk, ahol van internet hozzáférés, ott könyvelésünket és könyvelőnket azonnal elérjük. Adatainkhoz nem férhet hozzá senki (hacsak nem áruljuk el jelszavunkat), hiszen azok nem nálunk, hanem a NetLedger valamelyik központi szerverének adattárában húzódnak meg. Nem kell foglalkoznunk a könyvelőszoftver frissítésével, ezt a generál fejlesztő és üzemeltető automatikusan megteszi. Változnak a jogszabályok? Nem probléma, a szoftver azonnal igazodik hozzájuk. Automatikus audit, azaz hiteles kimutatás készül minden egyes tranzakcióról, törlésről. És arról is, hogy azt melyik felhasználó mikor és honnan kezdeményezte.

A szolgáltatást egyelőre csak magánszemélyek számára kínálja a NetLedger, de szerintem nem kell sokáig várniuk a kis cégeknek és középvállalkozásoknak sem. Az ügyviteli és számítástechnikai kapacitás csökkentésével is pénzt takaríthatnak meg. Kérdés: mi lesz az utcára került kontírozó könyvelőkkel? Bár, ahogy vannak utcai zenészek, miért ne lehetnének utcai könyvelők is? A kalapba dobott pénzről automatikusan bizonylatot, nyugtát állítanának ki. Az ügyesebbek számára máris nyilvánvaló: tisztára mosható a kalapba látványosan bedobott, sőt a be nem dobott pénz is. És még az áfája is visszaigényelhető. Feltaláltam tehát a mobil könyvelőirodát... A jogdíjakat nem a Keszóhoz, hanem a szerkesztőségbe kérem!

Herczeg József
herczeg@keszo.com

Netirodalom:

www.thinkfree.com
www.myfreedek.com
www.nuomedia.com
www.xenote.com
www.netledger.com

BeOS ikonok és ablakok

Testre szabható eszköztár

Előző számunkban ismertettük a BeOS installálásának menetét. A BeOS-hoz vezető úton azonban csak az első lépés a Personal Edition telepítése a sok esetben kényszerűségből használt Windows felületre. Ha tényleg megnyerte tetszésünket az új operációs rendszer, akkor a folytatásban túlléphetünk annak bizonyos korlátain, és átformálhatjuk azt saját ízlésünknek megfelelően.

Aki nem engedheti meg magának a BeOS kereskedelmi verziójának megvásárlását, az a Personal Edition változatot kézreállóbbá teheti néhány egyszerű trükk segítségével. Megszabadulhat még attól a korláttól is, hogy az 512 MB-os lemezterületen belül kell maradni. Kezdjük is ezzel. (Az alábbiakat természetesen számítógép előtt ülve érdemes kipróbálni.)

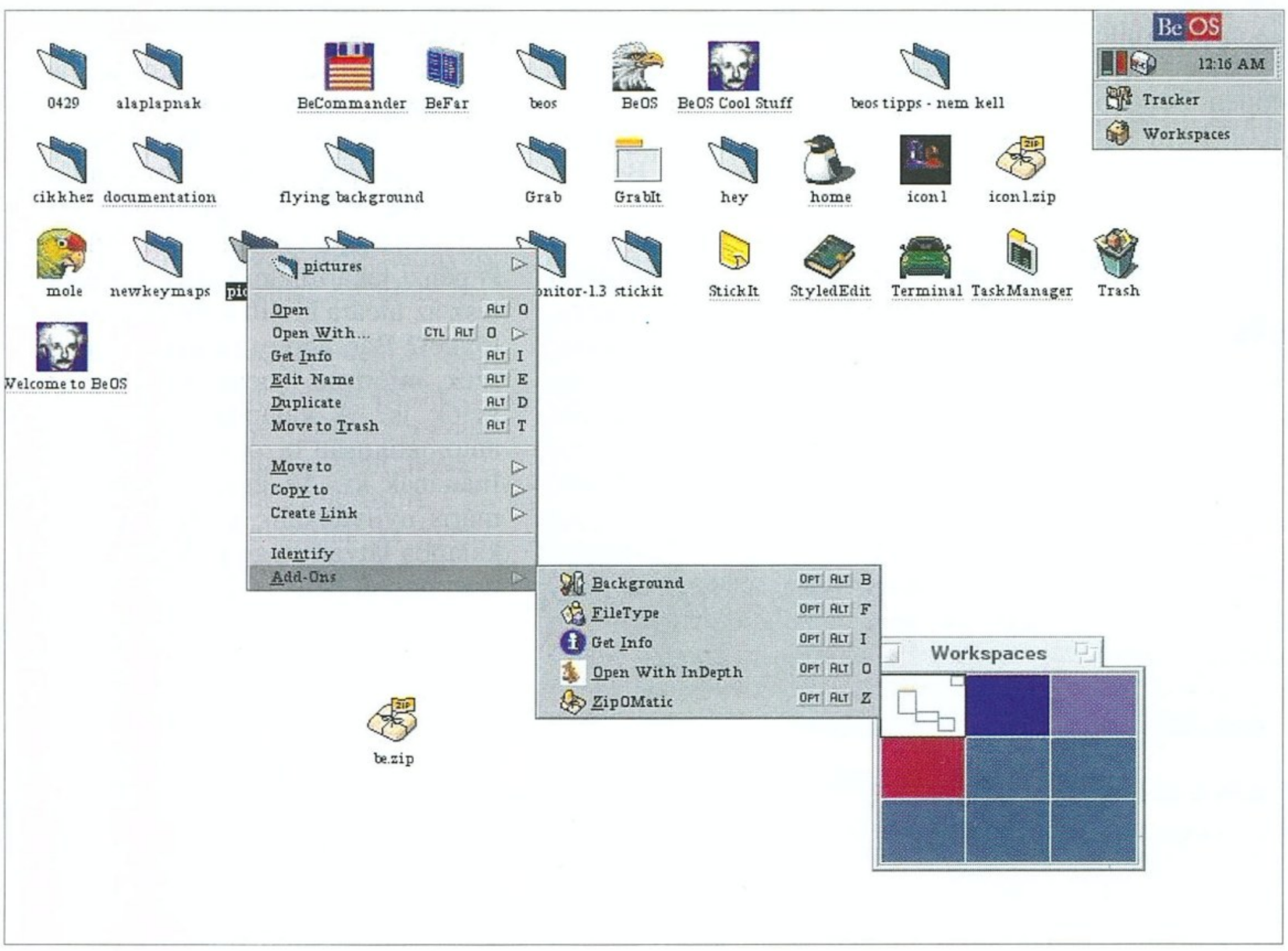
Túllépni az 512 MB-os korláton

Mint ismeretes, a Personal Edition telepítője alig 50 MB méretű, de a telepítés során 512 MB „működési területet" foglal le a megadott windowsos partíción. Ebből maga a BeOS operációs rendszer a beállításoktól függően igénybe vehet akár 200 MB-ot is, tehát nem túl sok hely marad a nagyobb alkalmazások telepítésére. A hely szűkössége zavaró, és sokan szívesen kiiktatnák a Windows társbérletet is. A Be FAQ ugyan azt állítja, hogy mindez csak a fizetős verzió beszerzésével oldható meg, valójában egy kis barkácsolással átléphetők a Personal Edition korlátai. Ezzel persze senkit nem akarok lebeszélni a kereskedelmi változatról, aki csak teheti, vegye meg a fizetős Be-t, hiszen az bőven megéri, különösen annak, aki tényleg dolgozni akar vele.

A továbbiakban két feltételezéssel élek. Az egyik az, hogy a cikk olvasója kezelni tudja az fdisk-et és a hasonló segédprogramokat, a másik, hogy valamilyen szinten már megismerkedett a Be-vel, tehát tisztában van például azzal, hogy a jobb egérgombbal egy alkalmazásra rákattintva, és az egérgombot folyamatosan lenyomva tartva, egy menü hívható elő.

Az első lépésben lemezformázó (fdisk) segédprogramunkkal hozzunk létre két partíciót. Az egyikre telepítsük a Windowst. A másikra majd a Windowstól függetlenül működő BeOS verzió fog kerülni. Azt is formázzuk azonban le, és ügyeljünk rá, hogy jó nagy legyen. Ezután telepíthetjük a Be-t a szokásos módon Windows alá. Ha készen vagyunk, indítsuk el a DriveSetup-ot. (Ha nem akarjuk, hogy folyton kergetesse a floppymeghajtónkat, akkor előtte tegyünk abba is valamilyen lemezt.)

A következő lépésben válasszuk ki a Be-nek szánt partíciót, de az ne legyen csatolva, mountolva. (A mountolás a fájlrendszerhez való hozzáfűzést jelenti, de talán egyszerűbb úgy értelmezni, hogy ami mountolva, csatolva van, azt ugyanúgy el tudjuk érni, mintha a merevlemezünkön lenne.)



A DriveSetup úgy érhető el, hogy a jobb felső sarokban található BeOS menü tetejére rákattintva előhívunk egy legördülő menüsort, és a Preferences részeként ott van a DriveSetup. Rá is kereshetünk persze a legördülő menüből (Find) avagy az Alt+F kombinációval. Itt jegyezzük meg, hogy a Be-nél alapértelmezésként általában Alt gombhoz kapcsolódó billentyűkombinációk vannak definiálva, szemben a windowsos Ctrl-kombinációkkal, de ezt át lehet állítani a Be menüből a Pre-

ferences, majd a Menu pontot választva, és mindenki maga eldöntheti, hogy megszokott beidegződéseinek megfelelő megoldást választ, vagy inkább valami újat, különbözőt.

Ha megvan a DriveSetup, akkor a jobb egérgombot tartsuk lenyomva, és úgy haladjunk végig a Setup — Initialize — BeOS (utána a partíció neve) — Be File System pontokon. Vigyázat! Ezzel természetesen minden adatot törlünk az adott partícióról!

Aztán az Applications alól az Installer következik: itt jelöljük meg, hogy hová (onto) szeretnénk telepíteni a Be-t, és indítsuk el a programot (begin). Nem árt tudni, hogy az Installer az egyik partícióról mindent átmásol a másikra, tehát ha már vannak dokumentumok, akkor azokat is. A folyamat lezárulása után indítandó el a Terminal (ez tulajdonképpen egy Bash shell), és — természetesen idézőjelek nélkül — gépeljük be, hogy „bootman", ami a boot manager rövidítése. A unixos hagyományoknak megfelelően itt nem mindegy, hogy „Bootman"-t vagy „bootman"-t írunk. Könnyű dolgunk lesz, ha minimális angol tudással rendelkezünk, mert egy ösztönösen kezelhető grafikus felület fog megjelenni. Itt egyelőre vegyük fel azt a partíciót is a boot menübe, amelyiken a Windows van, és azt is, amelyikre a Be rákerül.

Indítsuk újra a gépet, és a boot menüből válasszuk a nekünk tetsző operációs rendszert. A Be elindítása után egyébként értelemszerűen le is radírozhatjuk a Windowst, és átformázhatjuk azt a partíciót is BFS-re, azaz Be File Systemre, hogy még több helyünk legyen.

Csinosítgatás

A teljesség igénye nélkül lássunk most néhány olyan fogást, melyekkel a Be minden használójának érdemes megismerkednie.

Ikonok

Ha meg akarjuk változtatni egy alkalmazás vagy egy dokumentum ikonját (legyen ez a példa kedvéért a BeOS szövegszerkesztője, a StyleEdit), akkor először kattintsunk rá a jobb egérgombbal, majd a menü alján az Add-Ons pontból válasszuk a FileType-ot. Az ekkor megjelenő ablakban jobbra fent fogjuk látni jelenlegi ikonját (illetve ha nincs hozzá ikon társítva, akkor egy üres, szürke négyzetet). Most nyissuk meg ugyanígy azt is, amelyiknek az ikonját a StyleEdithez szeretnénk társítani. Az új képecskét egyszerűen vonszoljuk át (a bal egérgombot lenyomva tartva) a StyleEdit ikonablakába. Ha pedig azt akarjuk, hogy például az összes .html fájl ikonja változzon meg, akkor keressük meg azt az alkalmazói programot, amelyikhez a .html hozzá van rendelve (ez jelen esetben a NetPositive az Applications folderben). A Supported Types rovatban a fájl típusához rendelt ikont megváltoztathatjuk. Arra is van mód, hogy egy .jpg-ből vagy .bmp-ből (vagy más olyan képből, melyet ki tudunk nyitni Be alatt) ikont csináljunk. Ehhez az adott kép ikonját egy nyitott nyitott Icon-O-Matic, vagyis ikonszerkesztő ablakba kell áttenni. Ezt a legkönnyebben úgy tudjuk elindítani, hogy kétszer rákattintunk egy ikonra, vagy egy ikon üres helyére.

A Be-nél egyébként nincsenek „önállóan létező" ikonok, hanem azok a fájlattribútumokon „belül élnek". Ezért ha a sajátunkat másnak akarjuk odaadni, akkor először létre kell hozni egy üres fájlt (mondjuk egy üres StyleEdit dokumentumot), és ahhoz hozzárendelni az ikont. Utána célszerű tömörített formában továbbítani, mert a Be fájlrendszere gazdagabb, mint például DOS-é, ezért DOS-os lemezre másolva a benne lévő információk egy része elveszne. Ugyanez igaz általában is: ha becsatoljuk (mountoljuk) mondjuk a windowsos partíciót, akkor nyugodtan tárolhatunk fájlokat azon is, de csak zippelve. (A tömörítés a Zip-O-Maticnek köszönhetően Be alatt nagyon egyszerű: kijelöljük a tömörítendő fájlt vagy mappát, jobb egérgombbal előhívjuk a menüt, és az Add-Ons-ból a Zip-O-Matic-et választjuk.)

Az ablakok kerete

A BeOS menüt a Ctrl+Alt+Shift billentyűkombinációval kiválasztva egy egyébként rejtve maradó menüpont is megjelenik: a Window Decor, és innen akár az Amiga, akár a Macintosh, akár — horribile dictu — a Windows kinézetét választhatjuk. (És ugyanitt állíthatjuk vissza Be-re.) Szintén a kerettel kapcsolatos az is, hogy kétszer az ablak fülére kattintva az ablakot el tudjuk rejteni. Ha abban mondjuk egy StyleEdit dokumentum volt, akkor úgy tudjuk ismét előhívni, hogy a Be menüben a StyleEditre kattintunk, majd a megjelenő listáról kiválasztjuk. Ha jobb egérgombbal kattintunk a fülre, akkor az ablak nem tűnik el, csak a többi nyitott ablak mögé kerül. De azt is megtehetjük, hogy a Shift gombot lenyomva rákattintunk az ablak fülére, és ekkor a mozdulatlan ablakhoz képest tudjuk azt mozgatni, akár egészen a keret jobb széléig. A Preferences alatt található Menu segítségével pedig azt állíthatjuk be, hogy milyen legyen a beugró (pop-up) menü: milyen betűket, használjon, milyen legyen a színe stb. És még mindig a menü: az Alt+Esc lenyomásával kattintgatás helyett a kurzormozgatókkal tudunk lépkedni. A jobb oldali Ctrl+Shift lenyomása mellett jobb egérgombbal a Dekstopra kattintva (Clean-up All) pedig az összes ikont rendbe rakhatjuk.

Képernyőkép

A Print Screen hatására „screen1.tga" néven a /boot/home/ könyvtárba kerül a képernyőről készült „fénykép" (screenshot). Utána ezt megnyithatjuk, egérrel kijelölhetünk benne egy részt, ikont is csinálhatunk belőle, a desktopra tehetjük, elmenthetjük ppm, png, jpg vagy bmp formátumban...

Workspaces, a munkahely

A Be lehetővé teszi, hogy egynél több, egymástól független munkahelyen (workspaces) dolgozhassunk. Alapértelmezésben 9 ilyet használhatunk, de ezt a számot akár 32-re is feltornászhatjuk, ha a Preferences menü Screen menüpontjában (tehát nem a Workspaces-ben!) ezt beállítjuk. Legfeljebb 12 munkahely használata esetén az Alt+F1 ... Alt+F12 gombokkal navigálhatunk közöttük.

Ha ennél többel dolgozunk, akkor viszont magát a Workspaces Preferences-t kell elindítanunk. Ez kicsiben mutatja, hogy milyen munkahelyeink vannak, melyikben mi van a desktopon, és a megfelelő munkahely ablakára kell rákattintanunk.

Megtehetjük azt is, hogy a /boot /home/config/boot//UserBootscript-be beírjuk, hogy „boot/preferences /Workspaces &". Ezzel azt érjük el, hogy a Workspaces minden bootoláskor elinduljon. Az „&" lényegében azért kell, hogy a Bootscriptet anélkül folytassa az operációs rendszer, hogy várna a munkahely betöltődésére. Érdemes rá odafigyelni, hogy a betöltő szöveg neve eredetileg UserBootscript.sample, és el kell hagynunk belőle a „.sample" részt, ha azt akarjuk, hogy végrehajtódjon. (Ugyanez a helyzet a kilépésre szolgáló Shutdownscripttel is.)

Természetesen megtehetjük, hogy a különböző munkahelyeknek különböző háttérszínt választunk (jobb egérgombbal a desktopra, Add-Ons, Backgrounds). Vagy akár külön háttérképet is rendelhetünk mindegyikhez. Ha már itt tartunk: a háttérképet soha ne tartsuk a másik Be-partíción, mert azt az operációs rendszer csak a betöltődés befejezése után csatolja be (ha egyáltalán

úgy van beállítva, hogy automatikusan csatolja az összes Be jellegű partíciót). Azt is megtehetjük, hogy rákattintunk egy nyitott ablak fülére, majd lenyomjuk a kiválasztott munkahelynek megfelelő billentyűkombinációt (a 2. munkahely esetében Alt+F2). Persze átráncigálhatjuk a nyitott ablakot a munkahely felületén (mondjuk az 1-es mezőből a 3-asba), de eljuthatunk egy meghatározott munkahelyre a Terminalból is, elég annyit bebillentyűznünk a parancssorba, hogy Workspaces 3, és máris a 4-esen vagyunk (ugyanis a számozás 0-val kezdődik!).

Mindez azáltal válik igazán érdekessé, hogy innen kezdve az indítószövegbe (a bootscriptbe) beleírva is meghatározhatjuk a munkahelyeket, illetve arra is lehetőségünk van, hogy különböző munkahelyekhez különböző felbontásokat rendeljünk hozzá. Például egy készülő weblapot úgy is tesztelhetünk, hogy egyszerűen csak ablakról ablakra mozgatjuk, és megfigyeljük, hogy milyennek látszik különböző paraméterek esetén.

Az egér idomítása

Ha a Preferences menüben a Mouse-t választjuk, akkor érdekes lehet a „Focus follows mouse” pont alatt a „Warping” kijelölése. Ha ilyenkor az elrejtett ablakot a Trackeren keresztül előhívjuk, a kurzor az ablak közepére fog ugrani. Amikor pedig a kurzort a képernyőn mozgatjuk, mindig az az ablak válik aktívvá, amelyik éppen alatta van.

Twitcher

Ezt a funkciót a Ctrl+Tab együttes lenyomásával lehet előhívni (ha nem állítottuk át a billentyűkombinációkat), és nagyjából a Windows Alt+Tab funkciójának felel meg. A Twitcher megmutatja, hogy éppen milyen alkalmazások futnak, és a kurzormozgató billentyűkkel (vagy akár az egérrel is) átválthatunk egyikről a másikra, illetve az adott munkahelyeken futó alkalmazások esetében kiválaszthatjuk, hogy melyik dokumentum jelenjen meg a desktop tetején, a többi felett. Innen persze nem tudjuk kiirtani (kill) a programokat, de ha valami bajunk van velük, akkor a jól ismert Ctrl+Alt+Del billentyűkombinációt kell (egyszer) lenyomni. Ekkor az ún. Team Monitor jelenik meg, amely használható „Force Reboot” gyanánt (újraindítás mindenáron), akár arra, hogy kilőjük az okvetetlenkedő alkalmazást, akár pedig arra, hogy megnézzük: éppen milyen programok futnak.

Galántai Zoltán
galantai.zoltan@drotposta.hu

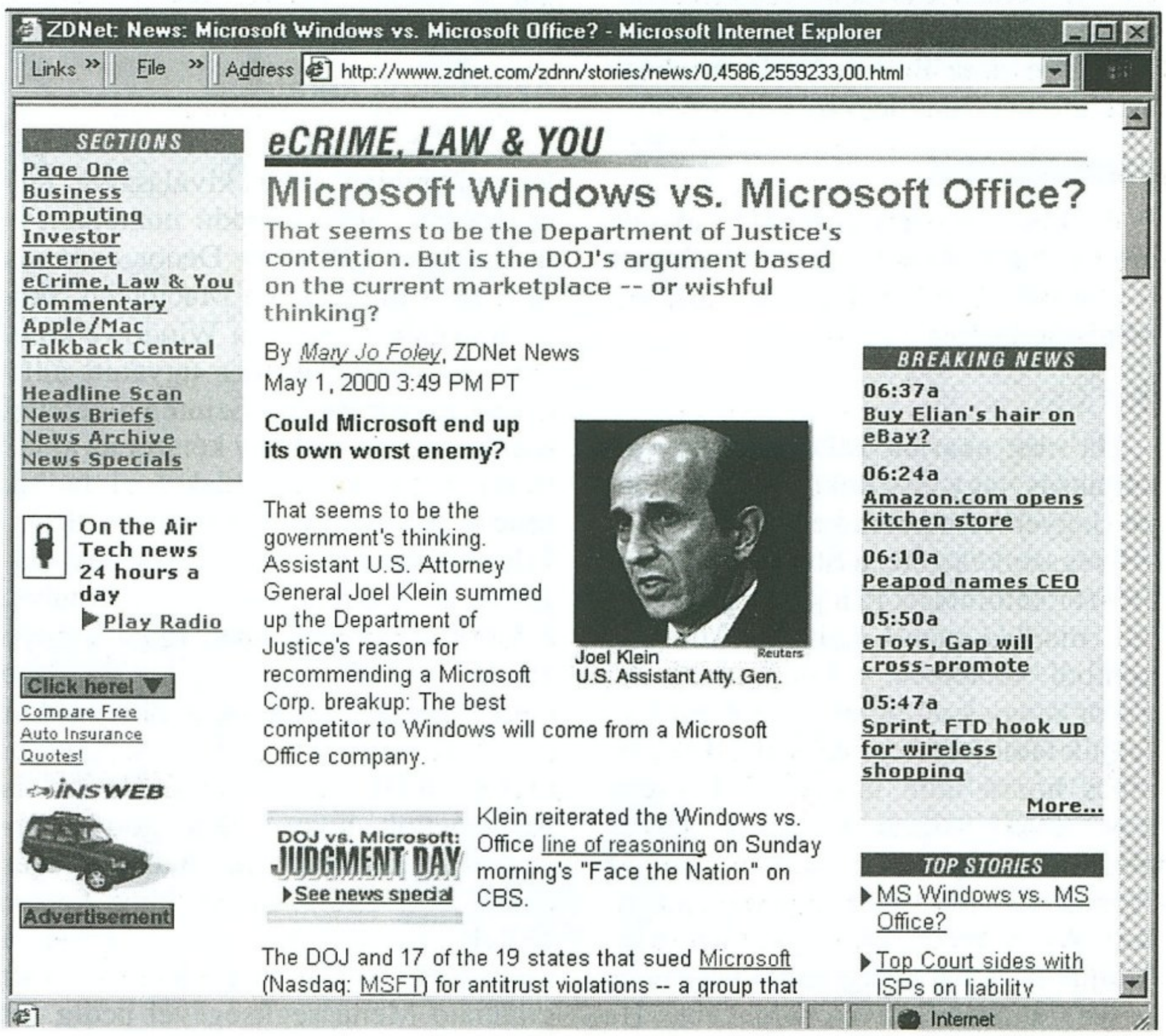
Windowstól a nyílt forráskódig

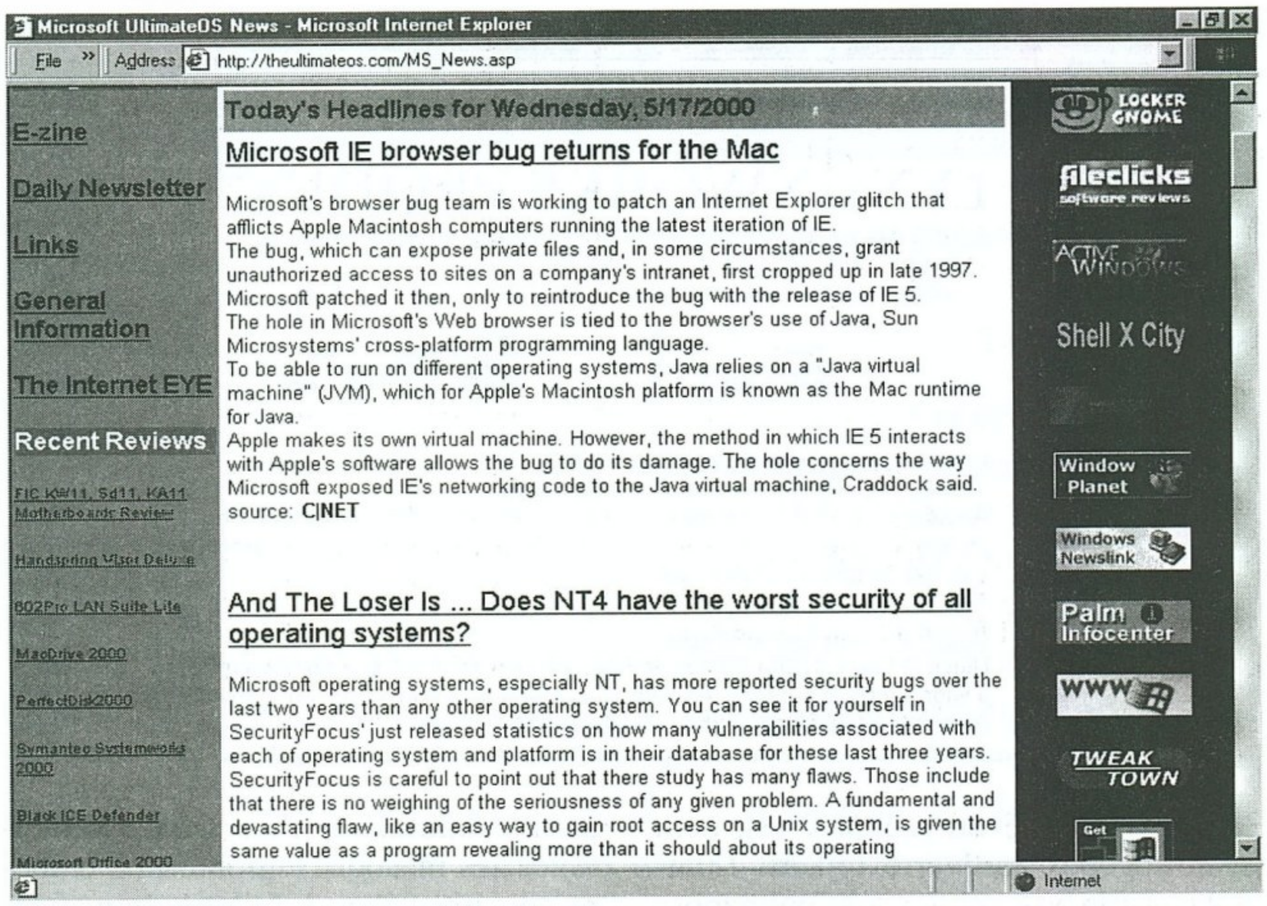
Ellenjavaslat

Miután a Microsoft ellen folyó perben Jackson bíró kimondta, hogy a cég törvénysértést követett el, most a szankciók megállapításának menete következik. A kormányzat szerint kétfelé kellene vágni a Microsoftot: az egyik az operációs rendszereket fejlesztené, a másik az alkalmazásokat. A javaslatot Bill Gates természetesen elfogadhatatlannak tartotta és ellenjavaslatot terjesztett elő, amely valamivel nagyobb mozgásteret biztosítana a konkurenseknek. Például igény szerint átalakítható lenne a Windows háttérképernyője, és „elrejtenék” az Internet Explorert. Ez viszont lényegében semmilyen „visszavonulást” nem jelentene, a böngésző továbbra is „be lenne drótozva” a rendszerbe, legfeljebb nem jelenne meg annyira tolakodóan. A szoftvercég elfogadna bizonyos adminisztratív intézkedéseket is, például a partneri szerződések módosítását. Szóba került még a Windows forráskódjának teljes és naprakész nyilvánosságra hozatala (legalább a többi szoftverfejlesztő cég számára), ami megkönnyítené a stabil és jól működő programok készítését. A Microsoft erre persze aligha lesz hajlandó, de félmegoldásként esetleg részletesebben közzéteszi a programozói felületeket, ha ezen az áron elkerülheti a feldarabolást. Az ellenajánlat nem hatotta meg a kormányzati szerveket, mert abban semmi nem garantálja, hogy a Microsoft a jövőben nem él vissza monopolhelyzetével, és az eddig okozott károk jóvátételéről is mélyen hallgat. A redmondiak tarsolyában azonban még maradt adu, persze nem technológiai, hanem pszichológiai: a felhasználók többsége ellenzi a felosztást. Ők ugyanis attól tartanak (és erre a Microsoft szívesen hivatkozik), hogy ez a lépés visszavetné a fejlődést, és végső soron a fogyasztók járnának rosszul. *(ZDNet)*

Whistler-változatok

A Microsoft a jelek szerint most már tényleg elhatározta, hogy a jövőben egységessé teszi a Windows 2000 (valójában NT 5) és a Windows 9x irányvonalat, és ezentúl ugyanarra a kernelre





épülő operációs rendszert kínálja a vásárlóinak két változatban. A Whistler kódnevű szoftver egyik verziója az üzleti szféra (business) számára készül, a másik otthoni használatra (consumer). Erről az úgynevezett következő generációs rendszerről bővebbet 1–1 1/2 év múlva tudhatunk meg, nem hivatalosan viszont a Whistler már két alkalommal is felkerült a hálózatra a Microsoft vezetőinek tudta és beleegyezése nélkül. Először a 2211.1 build számú változattal ismerkedhetett meg a világ, amelyet ugyan megpróbáltak az internetről leirtani, de a tükrözések sokkal gyorsabban jelentek meg, semhogy azt Redmondból követni tudták volna. Feltételezések szerint a tettes egy Microsoft alkalmazott volt, aki kimásolta a programot egy CD-re, majd feltette a webre. Minden bizonnyal ugyanennek a személynek köszönhető a 2223-as build kiszivárogtatása is, amely április közepe táján kezdett terjedni.

A hírverésre tehát nem lehet panasz, és a Microsoft illetékesei is elérkezettnek látták az időt, hogy nyilatkozzanak a Whistler tulajdonságairól. Szerintük az a cél, hogy a jóval megbízhatóbb Win2k kódjára alapozva akkora flexibilitást nyújtsanak, amekkorát a Windows 98 SE és a hamarosan megjelenő Windows Me biztosít. Például az üzleti szféra számára készülő változatban lenne egy PC Health nevű szolgáltatás, amely időről időre „pillanatképet" készítene a rendszerről, így rendszerösszeomlás esetén a visszaállítás egyszerűen megoldható. Az ígéretek szerint lehetőséget kínálnak az eszközmeghajtók (device drivers) automatikus frissítésére, több figyelmet fordítanak a mobil rendszerekre, és egyszerűbbé válik az internetes integráció is. A kezelhetőséget itt is varázslók (wizards) segítik majd, megújul a tálca (taskbar), és ennek köszönhetően elkerülhető lesz az ikonok túlburjánzása. A fogyasztói (consumer) változatról kevesebbet lehet hallani, annyit azonban sejthetünk, hogy felhasználói felülete egyszerűsödik, könnyen kiiktathatjuk majd a jelszavak használatát, és nagyfokú támogatást nyújt az otthoni hálózatok (home networks) kiépítéséhez és beállításához. *(PC World.com – ZDNet)*

Telepítőből „élesztő"

A PC-kre eladás előtt rendszerint telepítenek valamilyen operációs rendszert, az esetek döntő többségében egy Windows változatot, mellékelve hozzá telepítő CD-t is. Ez utóbbival a rendszeresen bekövetkező rendszerösszeomlások következményei újratelepítéssel elsimíthatók. A Microsoft véleménye szerint azonban ez a gyakorlat elősegíti az illegális másolatok terjedését, ezért a Windows 2000-rel útnak indított számítógépekhez ezentúl nem jár telepítőlemez, csupán egy úgynevezett „recovery CD", vagy a merevlemez külön partícióján elhelyezett „recovery image". Szükség esetén az operációs rendszer ezek valamelyikével helyreállítható, de kizárólag az eredeti telepítéskor érvényes hardverkonfiguráción. Tehát ha közben történt egy partícióváltoztatás vagy új hardverelemet helyeztek el a gépben, akkor a felhasználó pórul járhat. Sőt amikor változatlan marad a gépkonfiguráció, csak éppen a Win2k telepítésekor bizonyos elemekre nem volt szükség, később azt már sem a recovery CD-ről, sem a recovery image-ből nem lehet elővarázsolni. A hírek szerint ez a változás kiterjed a számítógéppel terjesztett összes Windows oprendszerre. Egyetlen kivétel a Windows 2000 Server Edition lesz. Ez utóbbihoz továbbra is hagyományos telepítő CD-t adnak, míg a többihez csak felélesztő (recovery) változatot. *(PC World.com)*

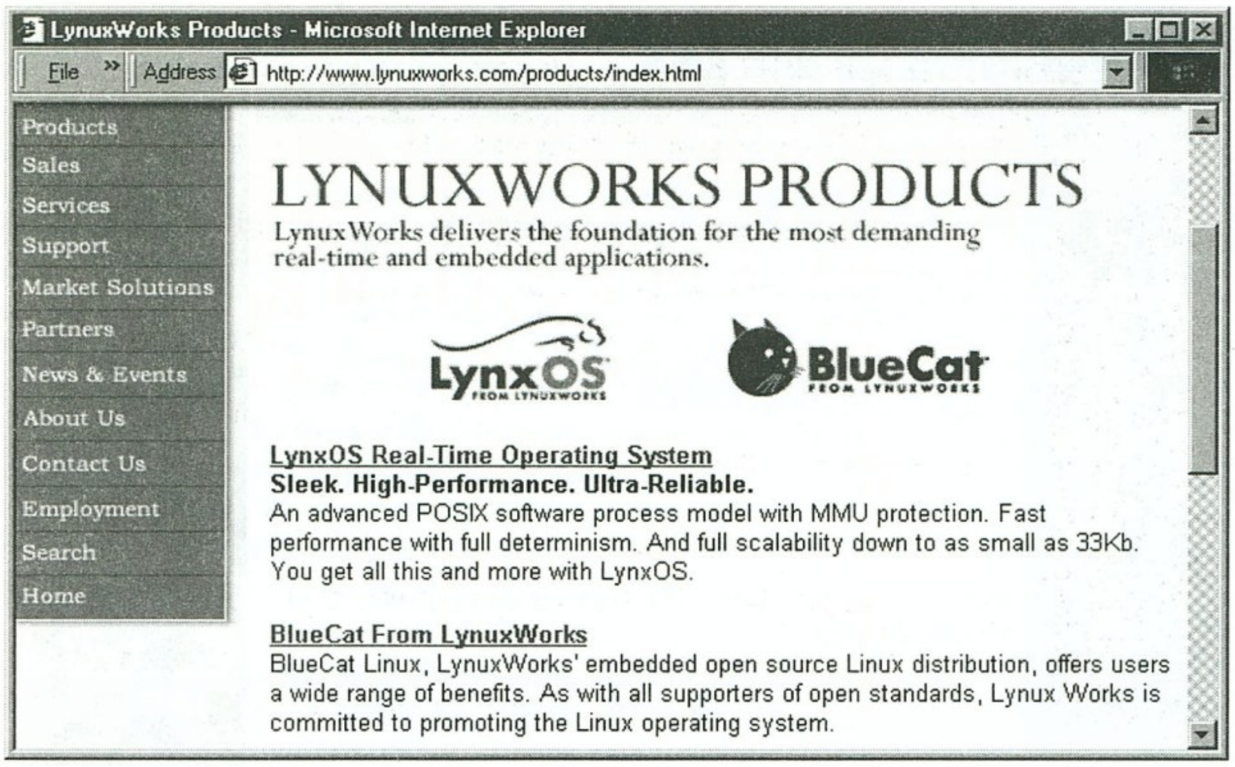
Biometrikus azonosítás

Az elektronikus kereskedelem fellendülése és az interneten keresztül lebonyolított sokféle ügyintézés (például szerződéskötés, adóbevallás, szavazás stb.) miatt kulcskérdéssé válik a személyazonosság megbízható megállapítása a cybertérben. Az eddig használatos felhasználói azonosítónév (login name) és jelszó (password) erre a feladatra nem igazán alkalmas. Az erős védelmet nyújtó véletlen karakterkombinációk szinte megjegyezhetetlenek, az egyszerűek viszont könnyen feltörhetők. A megoldást a jövőben a biometriai módszerek alkalmazása jelentheti, amikor azonosságunkat testünk valamelyik állandó megkülönböztető jegyével bizonyítjuk (ujjlenyomat, hang, írisz stb.)

A Microsoft nemrégiben megvásárolta az I/O Software cég Biometric Application Programming Interface (BAPI) elnevezésű technológiáját, valamint a SecureSite azonosításra használt eljárását. Ezek fogják a jövőben megjelenő Windows változatok biometriai alapon történő felhasználó-azonosító rendszereinek alapját képezni. A biometriai megoldások idővel háttérbe szoríthatják a jelszavak használatát. A BAPI és a SecureSuite lehetővé teszi az emberek fizikai jellemzőire vonatkozó adatok tárolását, az azonosító rendszerek működtetését, ugyanakkor egységes felhasználói felületet kínál a biometriai azonosító módszereket támogató programok számára, egyelőre még csak Windows 9x és NT 4.0 platformon, de a tervek szerint ezen a nyáron megjelenik a termék Windows 2000-es változata is. *(PC World.com)*

Beágyazott Linux

A Linux várhatóan a beépített alkalmazásokban is jelen lesz, a konyhától a garázsig mindenütt. A tervek szerint a Motorola és a TurboLinux érdekeltséget szerez a Lynx Real-Time Systemsben, mely éppen lapzártakor változtatta meg a nevét. A továbbiakban



LynuxWorks néven működő cég nem számítógépes hardverre fejleszt Linuxot (set-top box, printer, telefonközpont stb.), vagyis olyan „beágyazott" operációs rendszereket ír, amelyek közvetlenül a mikroprocesszorra vannak ráültetve. Ma a „láthatatlan számítógépek" (üzemanyagpumpák az autókban, repülőgépes navigációs rendszerek stb.) 90%-a beágyazott operációs rendszert használ. Ráadásul ezen a területen a Microsoft nem tudott monopolhelyzetbe kerülni. Mivel a Linux kicsi, stabil és gyors (illetve modularitása miatt „testre szabható"), esélyesebbnek tűnik, mint a Windows CE. Ráadásul a forgalmazónak nem kell a végtermékért licencdíjat fizetnie. Paul Czerny (Venture Development Corporation) úgy véli, hogy mivel beágyazott operációs rendszert sok és olcsó berendezésben használnak, ezért „néhány fillér is sokat számít. Ha Windows CE-t használnánk, az tízszer drágább lenne." A szakértők biztosra veszik, hogy a jövő azoké a berendezéseké, melyek egyrészt a webre tudnak kapcsolódni, másrészt az egész lakásban képesek egymással kommunikálni. Az IDC szerint 2004-ben 89 millió ilyen berendezést fognak eladni. A beépített operációs rendszerek szintjén a jövőben a Windows CE mellett a Linux vetélytársa lehet a BeIA (a BeOS fejlesztésére alapozva) és a mikrokerneles QNX (a Unix egyik változata). *(Yahoo!News – ABCNEWS.com)*

Új kernelre várva

A Linux 2.4-es verziószámú következő kernele Linus Torvalds szerint augusztusra vagy szeptemberre várható (vagyis 18 hónappal a 2.2-es kernel kibocsátása után), bár az eredeti tervek szerint már 1999-ben le kellett volna zárni a fejlesztését. Alan Cox, a kernelfejlesztés második legfontosabb embere egy interjúban kifejtette, hogy „inkább engem bántson a késés, mint a felhasználókat a nem megfelelően megírt kód". A felelősséget az is növeli, hogy egyre több nagy cég áll át Linuxra. Torvalds szerint egyébként eredetileg nem is volt határidő kitűzve, csupán „mindenki tudta, hogy a 2.0-s és a 2.2-es között eltelt 2 és fél év túlságosan sok". Az új, 2.4-es kernel egyik legfontosabb jellemzője lesz, hogy magas szinten fogja támogatni a multiprocesszoros rendszereket (az SMP-t, vagyis a szimmetrikus multiprocesszoros megoldásokat), és ennek megfelelően minden eddiginél alkalmasabb lesz arra, hogy nagy teljesítményű szervereken fusson. A késlekedését egyébként részben az okozza, hogy Torvalds szerint az új fejlesztésekkel járó hibákat „jobb előbb, mint később kijavítani", ezért már februárban felszólította a programozókat, hogy inkább a hibakeresésre koncentráljanak a további fejlesztések helyett. A 2.3.0-s (vagyis kizárólag fejlesztőknek szánt, páratlan sorszámú) kernel már korábban megjelent, nemrégiben pedig kibocsátották a 2.3.99-eset is. Az 1999 januárjában közreadott 2.2-es kernel a 4 processzoros gépek teljesítményét is ki tudta használni, a mostani kernelt pedig 8 processzoros komputerekre is optimalizálni fogják. A jövőben akár a 64 processzoros szupergépeken (például Sun E10000-es szerveren) is futtatható lesz a Linux. A késedelem egyébként azzal is magyarázható, hogy a kernel fejlesztésében részt vevő programozók közül kevesen dolgoznak otthon kétprocesszoros gépekkel, a 8 processzoros gépek vásárlása pedig végképp nem jellemző. *(CNET.com – ZDNet)*

A Red Hat a legjobb

A hálózatokkal foglalkozó mintegy 200 000 szakemberhez eljutó Network Magazine idén 13. alkalommal hirdetett győztest a legjobb operációs rendszerek között. A szerverszoftverek közül ezúttal a RedHat 6.1-es verziója került az első helyre. A szabályok értelmében a zsűrizett terméknek mindig az előző évben kell megjelenni a piacon, így ugyanaz a termék kétszer nem indulhat. A minősítés sok más kategóriára is kiterjedt, a fontosabbakban az alábbi termékek kerültek az első helyre:

- **Tivoli Systems Storage Manager 3.7**
- **WebSphere 3.0 (IBM)**
- **NDS eDirectory (Novell)**
- **Gauntlet Firewall 5.5 (Network Associates)**
- **ClearTrust SecureControl 3.5 (Securant Technologies)**
- **Norton Antivirus Corporate Edition 7.0 (Symantec)**
- **Integrated Messaging Exchange (Tumbleweed Communications)**
- **MeetingPoint 4.0 (White Pine Software)**
- **LANMeter 686 (Fluke Enterprise)**
- **Bluetooth (Ericsson)**

(CMP Media – NewsAlert)

Postwindows korszak?

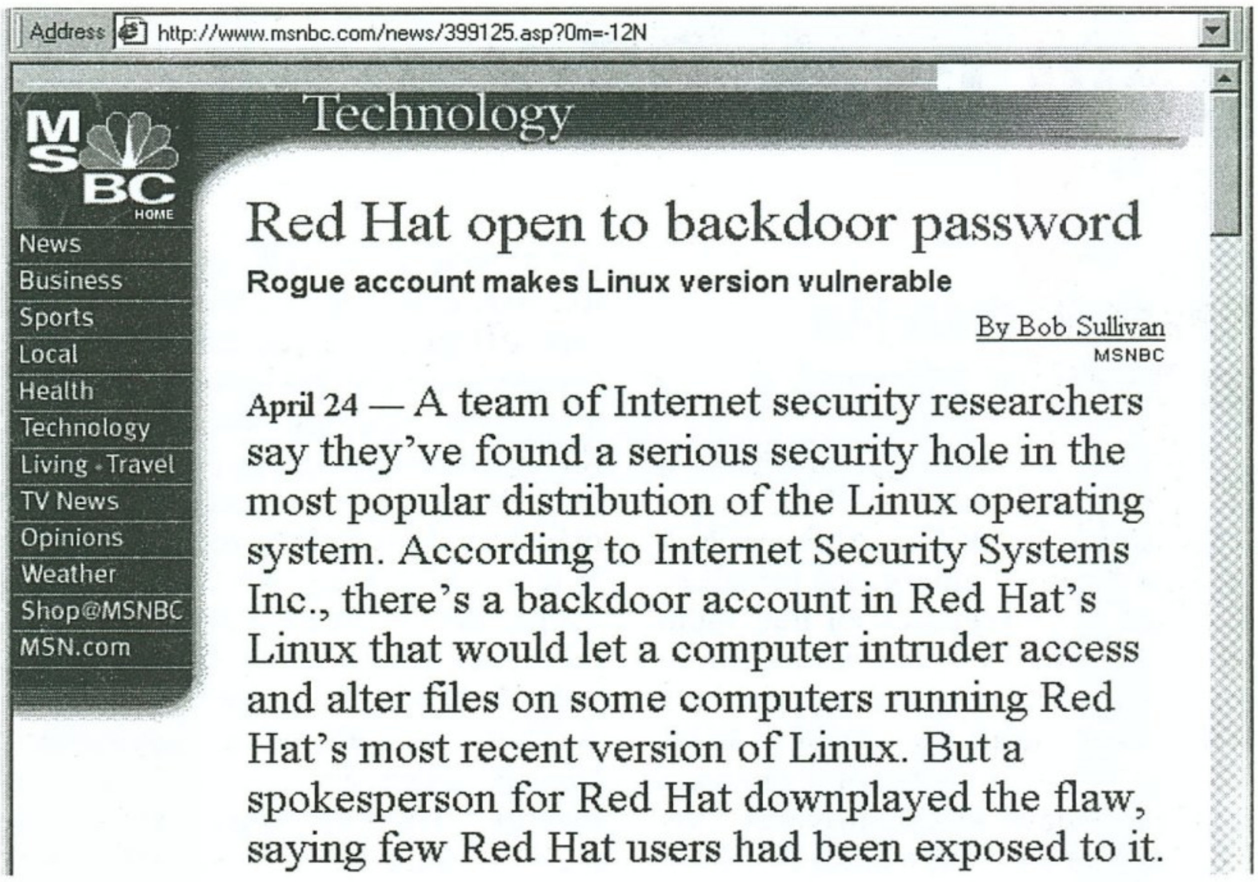
Miközben úgy tűnik, hogy pénzügyi (értsd „spekulációs" — A szerk.) körökben a „Linux-hisztéria" egyre inkább lecseng, a nyílt forráskód egyre inkább előretör az interneten. Az International Data Corporation (IDC) egyik legutóbbi felmérése szerint a Linux a leggyorsabban hódító operációs rendszer, és Michelle Bailey kutatási vezető szerint a Linux-szal összefüggő eladások „erősen kötődnek az internethez". Az operációs rendszerek piacán második helyen álló Linuxból 1999-ben 5,4 millió példányt adtak el. Ez persze jóval elmarad a Windows NT eladási számai mögött, és az előrejelzések szerint csak 2002-re fogja utolérni (jelenleg évi 166%-os a növekedés). A feltételezések szerint rövid távon nem sok változás várható, de ha bekövetkezne a Microsoft feldarabolása, akkor sokkal élesebbé válna a verseny.

Ha ugyanis sok driver és alkalmazás jelenne meg egyéb operációs rendszerekre, akkor a Microsoft operációs rendszerrel foglalkozó „utódának" is több innovációt kellene belvinnie a fejlesztésbe (ezenkívül egy vagy két „Baby Bill" kialakulása várható az Office fejlesztésekre és az MSN-re). Más

szakértők úgy ítélik meg, hogy csak a Windows forráskódjának szabaddá tétele adna igazi lökést a fejlesztésnek.

Mivel az utóbbi időben a számítástechnika „frontvonala" az internet volt, sokan úgy vélik, hogy bár Redmond az operációs rendszerek terén toronymagasan vezet, elkezdődött a „postwindows" korszak, hiszen a hálón a Microsoft fölénye nem olyan nagy, mint az asztali gépek kategóriájában, és egyre több nagy cég támaszkodik Linuxra. Az IBM például a Red Hat-et és a Turbo Linuxot támogatja, a Compaq pedig 1999-ben a Linux szerverek első számú forgalmazójává vált 84 millió dolláros bevétellel. Az IDC szerint még az IBM, a Dell, a Hewlett-Packard és a Fujitsu van rajta a Linux szerverek forgalmazóinak top 5-ös listáján.

Egyes információk szerint hosszú távon gondok lehetnek a Windows kernelének fejlesztésével is, mert az utóbbi időben számos szakember vált ki a Microsoftból, hogy saját céget alapítson. *(E-Commerce Times)*



Red Hat open to backdoor password

Rogue account makes Linux version vulnerable

By Bob Sullivan
MSNBC

April 24 — A team of Internet security researchers say they've found a serious security hole in the most popular distribution of the Linux operating system. According to Internet Security Systems Inc., there's a backdoor account in Red Hat's Linux that would let a computer intruder access and alter files on some computers running Red Hat's most recent version of Linux. But a spokesperson for Red Hat downplayed the flaw, saying few Red Hat users had been exposed to it.

Az ILOVEYOU tanulságai

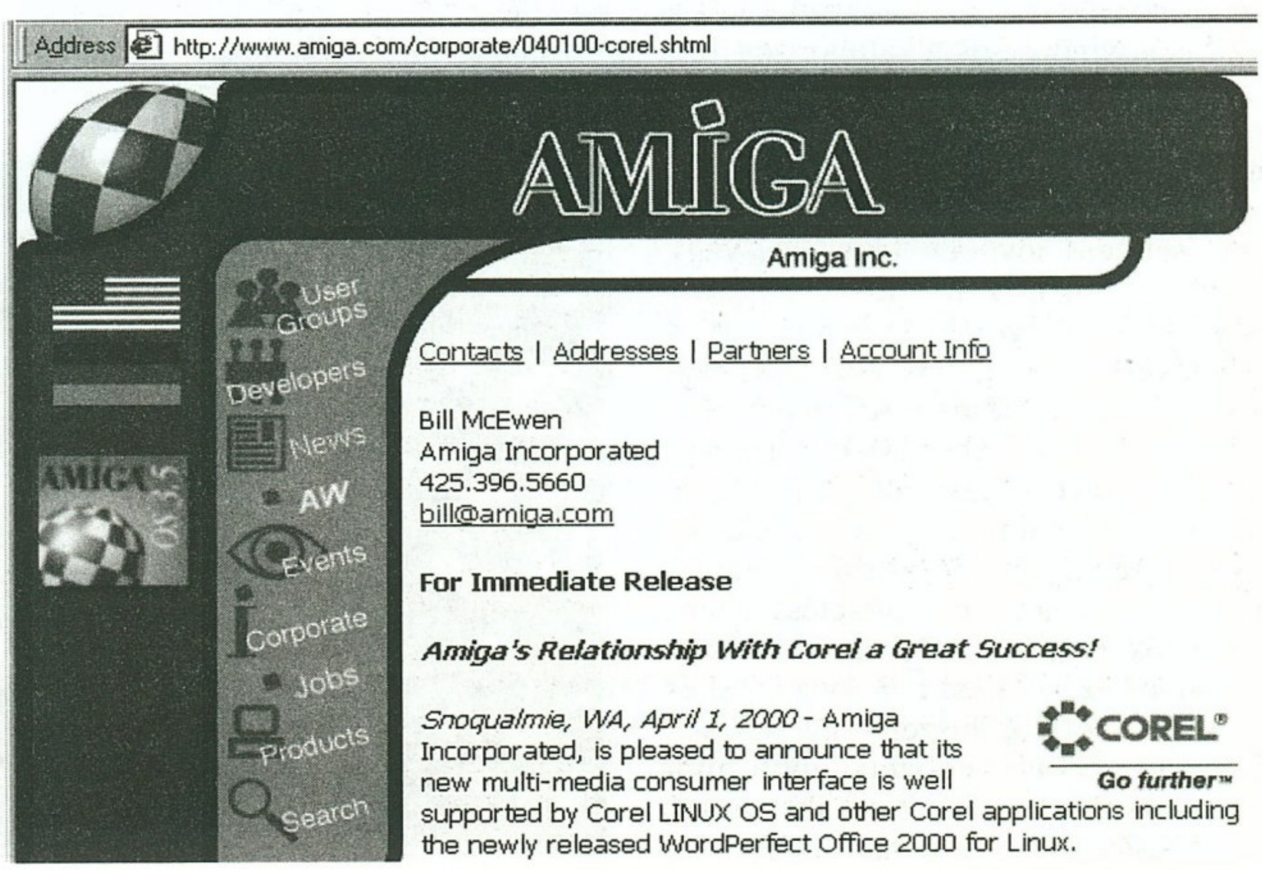
Az utóbbi idők leghíresebb rosszindulatú féregvírusa, az ILOVEYOU megfertőzte többek között az Egyesült Államok Védelmi Minisztériumának (Department of Defense), a Szenátusnak, sőt még a biztonság legfőbb őrének, az NSA-nak (National Security Agency) a gépeit is. Ez az esemény több kérdést is felvet. Egyfelől azt, hogy a legmagasabb biztonsági fokozatú helyeken hogyan engedhették meg a Windows és a biztonsági réseiről hírhedt Outlook levelező használatát ahelyett, hogy valami biztonságos megoldást választottak volna. Másfelől pedig olyan értelemben is felvetődik a kérdés, hogy ilyenkor mekkora a szoftver készítőjének, jelen esetben a Microsoftnak a felelőssége.

Ami a „jogot" illeti, a Licence Agreement szerint természetesen semekkora. Józan ésszel nehéz belátni, hogy következmények nélkül maradhat, ha egy cég ekkora védelmi rést hagy a rendszerben. Másfelől a nyílt forráskódú szoftverek használóinak helyzete sem sokkal jobb. Igaz ugyan, hogy a fertőzés elkerülte őket, de egy esetleges hasonló katasztrófa esetén a nyílt forráskódú szoftverek gyártói sem vállalnak semmiféle felelősséget, pedig sok esetben (Red Hat, OpenLinux, SuSE) már arra sem lehet hivatkozni, hogy a programozók hobbyból, ráérő idejükben fejlesztik a Linuxot. Hosszú távon a számítástechnika (elsősorban a beágyazott alkalmazások) rohamos terjedésével egyre fontosabb lesz, hogy a gyártóknak ugyanúgy vállalniuk kell a felelősséget a szoftverek hibás működéséért, mint egy autógyárnak vagy egy építőipari cégnek. *(Linux.com)*

Az Amiga románca

Az Amiga azt tervezi, hogy az Amiga interfészét a Corel Linuxon és Red Hat-en is futtathatóvá teszi, ugyanekkor a Sun azon dolgozik, hogy Amigára optimalizált Java Virtual Machine-t fejlesszen ki. Bill McEwen, az Amiga egyik vezetője szerint legalább 117, különböző gyártóktól származó alkalmazás — többségében játék — fog majd az új Amiga rendszeren futni. Egyes források szerint az új Amiga még idén nyáron megjelenik, és igazi valós idejű multimédia RTOS lesz (real-time operating system). „Amit a DirectX jelent a Windowsnak, ugyanazt jelenti majd az Amiga a Linuxnak", mondta McEwen az Amiga 2000 fejlesztői konferencián. Az Amiga a jövőben a mobiltelefonok és a kézi számítógépek gyártóinak Amigát futtató hardvereit is árulni fogja 600 amerikai boltjában. Ugyanekkor a Gateway az America Online-nal közösen akar 500 dollár alatti, Linux alapú internetes berendezéseket fejleszteni, ezek várhatóan az év végén kerülnek forgalomba, és fel lesznek szerelve Instant AOL-lal is. A fejlesztők reményei szerint ez a konfiguráció a PC-k komoly vetélytársa lehet. *(ZDNet)*

Galántai Zoltán – Mákos András



Nem csak OS/2 ...

Novemberben jön az OS/2-es kliens!

Most már biztos, hogy ismét lesz új OS/2-es kliens, mivel az IBM hivatalosan is bejelentette Convenience Pack (komfortcsomag) elnevezésű termékét. Mint azt a híresztelésekből már tudni lehetett, az új 4.5-ös verziószámú kliens a Warp Server for e-Business alapjaira épül, és az újítások egy részét az IBM-től már most is letölthető ingyenes frissítések (Java motor, driverek, fixpak, Netscape stb.) teszik ki. Szerepelnek azonban a listán olyan komponensek is (például 32 bites TCP/IP alkalmazások), amelyeket eddig fizetés ellenében lehetett letölteni, illetve olyanok (például a Logical Volume Manager), amelyekkel eddig csak a legújabb Warp szervert futtatók rendelkeztek.

A Convenience Pack rejteget tehát igazi meglepetéseket, ám a termék elsődleges célja tagadhatatlanul az, hogy megkönnyítse, kényelmesebbé tegye az OS/2-t telepítők és használók életét (erre utal az elnevezés is). Az IBM dolga is egyszerűbbé válik, mert ezzel a húzással közös alapokra helyezik a szervert és a klienst, így a jövőben nem kell kétfajta javítócsomagot kiadni. A Convenience Pack tartalmaz még egy igen jelentős változtatást, amelyet „elfelejtettek" bejelenteni a hivatalos közleményben. A Warp 4.5 virtuális DOS környezetében ugyanis megteremtik a 32 bites windowsos alkalmazások futtatásának elvi lehetőségét a VDM korábbi 512 MB-os címzési korlátozásának 2 GB-ra történő kitolásával. Ez igen jelentős lépés, mert a Win-OS2 környezet esetleges jövőbeli frissítése, vagy az Odin projekt konverterének (http://www.netlabs.org/odin/) telepítése után lehetővé válik a legtöbb Win32-es program futtatása OS/2 alatt. A Convenience Pack 2000. november 30-án fog megjelenni, és azt a regisztrált Warp 4 programmal, valamint a Software Choice vagy a Software Subscription under Passport Advantage előfizetéssel rendelkezők kapják majd meg.

Sajnos ez azt jelenti, hogy a frissítés nem lesz olcsó, hiszen egy kétéves Software Choice előfizetés több mint 200 dollárba kerül. Az egyik magyar számítástechnikai cég már elkezdte az érdeklődők összegyűjtését, mert remélik, hogy nagy tételben történő rendelés esetén ki tudnak majd csikarni bizonyos árengedményt az IBM-től. A részleteket a magyar OS/2-es levelezési listán (os2@lists.bme.hu) lehet megtudakolni. A dolog árnyékos oldalához tartozik még, hogy bár más nyelvek támogatását is tervezik, a Warp 4.5 egyelőre csak angol, német, spanyol, japán, kínai, portugál, dán, holland, finn, norvég, svéd, francia és olasz nyelven „beszél". A kliens megjelenésével egyidejűleg frissítik a szervert is, a jövőben pedig további komfortcsomagokat adnak majd ki. A kizárólag CD-ROM-on terjesztett terméket a megjelenést követő év december 31-ig támogatja az IBM.

Elkészült a FixPak 13

Akinek nincs pénze Software Choice előfizetésre, esetleg csak türelmetlen (nem tudja kivárni a november 30-át), az már most is elkezdheti használni a „Warp 4.5-öt", amennyiben letölti és telepíti a FixPak 13-at. A viszonylag hosszú szünet után kiadott javítócsomag ugyanis az egész alaprendszert lecseréli a Warp Server for e-Businessben találhatóra. Ez a javítócsomag fog a Warp 4.5 kliens alapjául is szolgálni. Természetesen a FixPak 13 telepítésével kapott új Warp a végleges változat által nyújtandó legtöbb funkcióval még nem rendelkezik, de például feloldja a VDM-ben található memóriacímzési korlátozást (lásd az az előző hírben is).

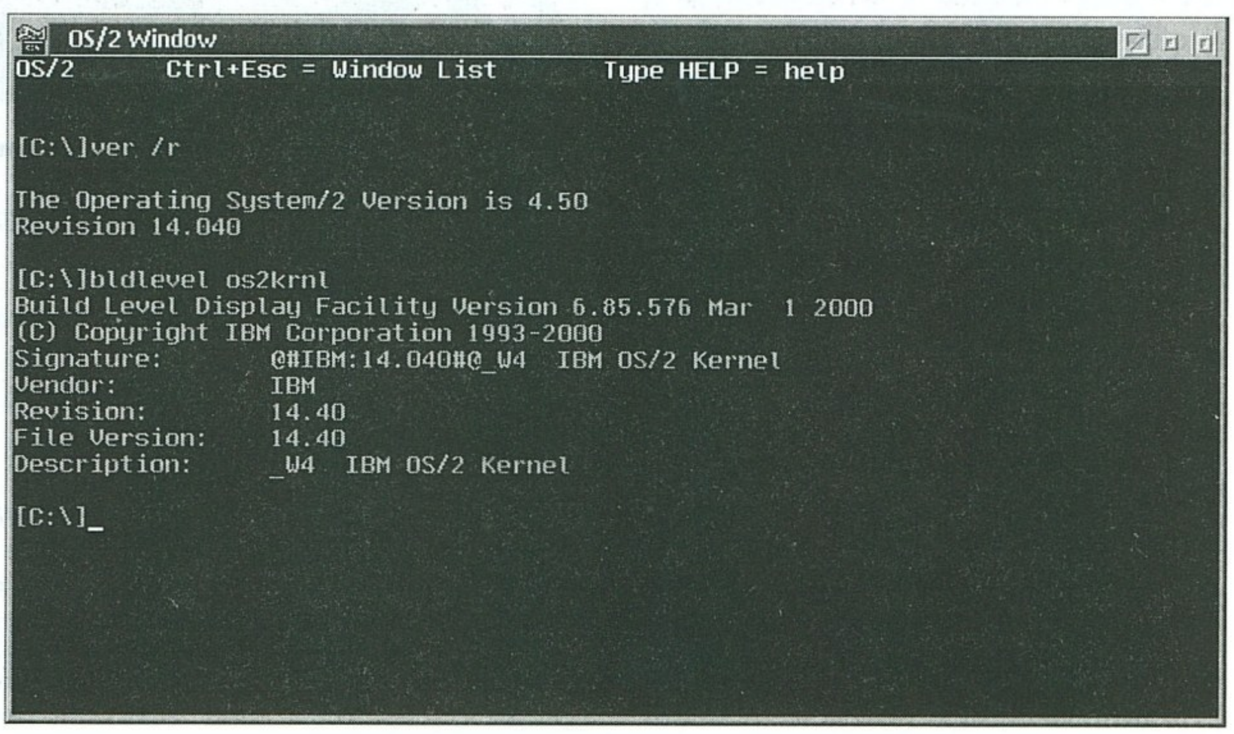
Bár a javítócsomagok az esetek többségében nem okoznak nagyobb problémákat, a felhasználók internetes visszajelzései alapján arra lehet következtetni, hogy a szokásosnál több a kellemetlenség ezen FixPak telepítése után. Az IBM emiatt már több kisebb javítást is kiadott, és tervezik a FixPak 13 javított változatát. Az egyelőre még csak angol nyelven kiadott javítócsomag az ftp://ftp.software.ibm.com/ps/products/os2/fixes/v4warp/english-us/xr_m013/ könyvtárban található, telepítéséhez legalább 1.42-es verziójú telepítőprogramra van szükség (ftp://ftp.software.ibm.com/ps/products/os2/fixtool/english-us/).

Java-tervek

Az IBM szorgalmasan dolgozik igen gyors Java motorjainak újabb változatain. A legfrissebb tervek szerint még ez év első felében elkészülnek a támogatott platformokra (AIX, Linux, OS/2, OS/390, OS/400, Windows) a Java 2-es motorok, melyekből jelenleg csak az AIX és az OS/400-as verziók érhetők el. A linuxos, OS/2-es és OS/390-es motorokból előzetes verzió (preview) kibocsátását is tervezik. Az 1.1.8-as motorokat továbbra is támogatja az IBM, bár a frissítések kibocsátásának ütemét a Sun által kiadott javítások frekvenciája határozza majd meg.

DB2 v7.1

A DB2 adatbáziskezelő 7.1-es változata még 2000 első felében megjelenik az eddigi platformokra (AIX, Solaris, Linux, Windows NT/2000, OS/2 stb.). A http://www-4.ibm.com/software/data/launch/ címen található híradás megnyugtatás azon operációs rendszerek használói számára is, amelyekre eddig nem készült a 7-es sorozatból béta verzió. A 7. generációs adatbázist már



teljesen az e-business jegyében készítették, például nagy gondot fordítottak a webkiszolgálókkal történő integrálásra. Az IBM nem rejti véka alá, hogy tovább szeretné erősíteni a relációs adatbázisok piacán kivívott előkelő pozícióját, mivel olyan eszközöket fog mellékelni a 7-es verzióhoz, amelyek lehetővé teszik a már meglévő adatbázisok konvertálását a DB2-nek megfelelő formátumra. A híradás szerint a termék árfekvése is kedvező lesz.

Driverfrissítések

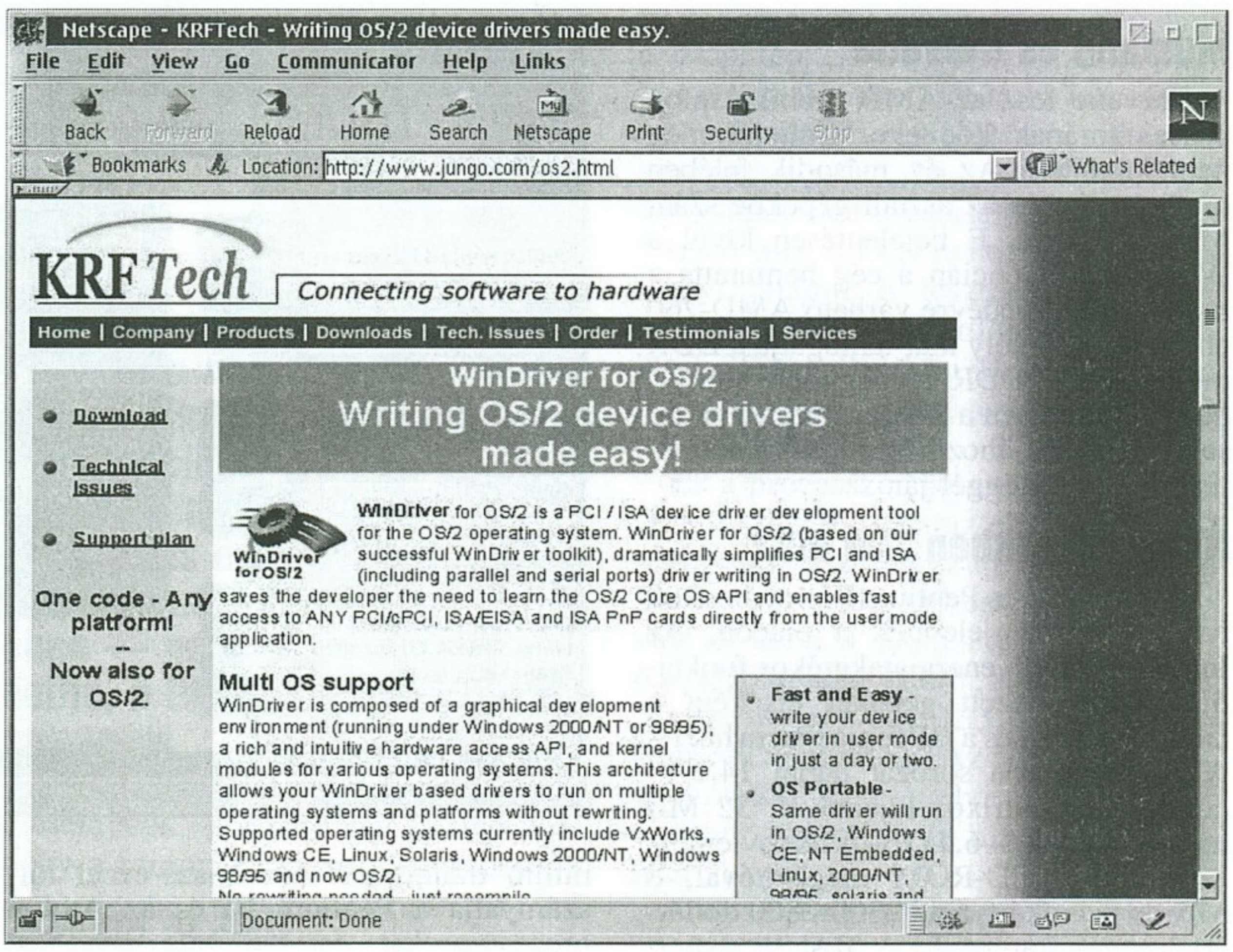
Az IBM továbbra is rendszeresen frissíti több ezer drivert tartalmazó webhelyét (OS/2 Device Driver Pak On-Line). Az elmúlt időszakban főleg a nyomtatók terén történtek lényeges változtatások. Megoldották például az USB nyomtatók támogatását (http://service.software.ibm.com/os2ddpak/html/6218FF37981954BC872568B90069B9B6.html), valamint frissítették a többnyire magánfelhasználó által birtokolt Omni (Epson, HP, Canon stb.) és postscript printerek drivereit (ftp://ftp.software.ibm.com/ps/products/os2/printpak/) is.

Flash for OS/2

Az InnoTek Systemberatung jóvoltából elkészült az első olyan Netscape Communicator bővítőmodul, amelynek segítségével megtekinthetők a flash technológiával készült weblapok. Az eredetileg a Macromedia által kifejlesztett flash technológia elsősorban azok számára lehet érdekes, akik igényes kinézetű, grafikával és hanganyaggal gazdagon ellátott interaktív weboldalakat akarnak készíteni, s mindezt olyan tömören, hogy rövid idő alatt le is lehessen tölteni. A jelenleg még béta verziójú, időkorlátos, ám egyes felhasználók szerint kifogástalanul működő bővítőmodul a http://www.innotek.de/flash oldalról szerezhető be.

Stellar Frontier

A többek között a Galactic Civilization játékáról ismert Stardock ingyenessé tette egy másik programját, a Stellar Frontiert. A játék az űrben zajló stratégiai játékok körébe tartozik. A Stellar Frontiert akár 64-en is játszhatják egyszerre a hálózaton keresztül. Bár az alkalmazás teljesen ingyenes, a fejlesztők azt kérik, hogy az érdeklődők lépjenek be az Admirálok klubjába, hogy ezzel is támogassák a továbbfejlesztést. Ha valaki kedvet kap hozzá, akkor a http://www.stardock.com/products/sf címen olvashat a belépés módjáról, és innen szerezheti be saját példányát is.



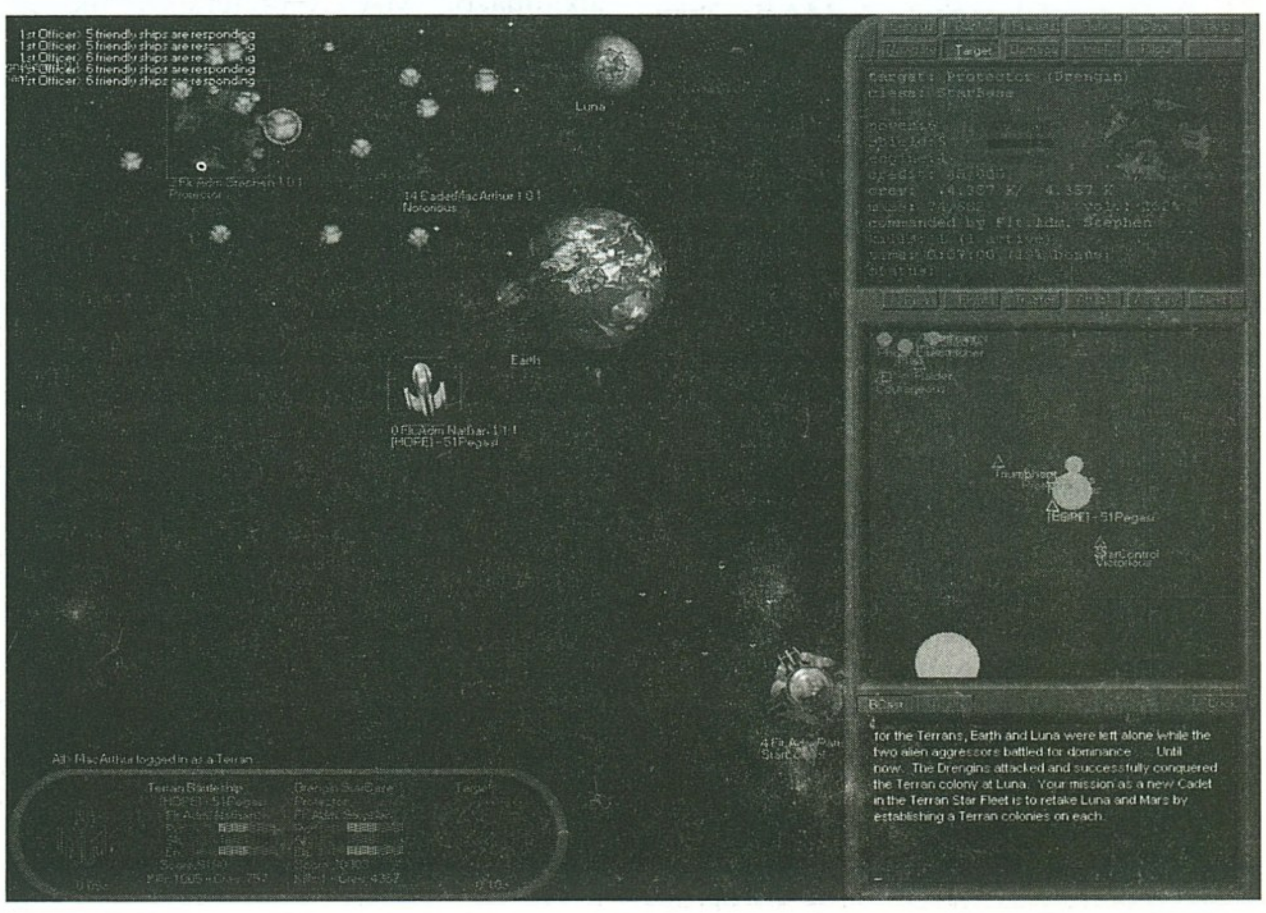
Nincs drivere OS/2-höz? Készítsen egyet a WinDriverrel!

WinDriver for OS/2

A KRFtech cég elérhetővé tette a driverek készítését lényegesen megkönnyítő WinDriver fejlesztőkörnyezet OS/2-es drivereket is támogató változatát. A WinDriver nagyon leegyszerűsíti a multiplatformos driverek fejlesztését, mert a kód birtokában a legtöbb esetben csak egy újrafordításra van szükség, hogy előállítsuk a különböző Windows, Linux, Solaris vagy most már OS/2 platformokon is használható drivert.

A PCI eszközöket támogató beépített könyvtárak segítségével sok esetben csak a támogatandó hardverelemet kell gépünkre telepíteni, és a kódgeneráló varázsló segítségével majdnem teljesen automatikusan lehet elkészíttetni a kívánt drivert. A jelenleg csak 32 bites Windows alatt futtatható, még béta állapotban lévő program a http://www.jungo.com/os2.html oldalon keresztül tölthető le.

Kádár Zsolt



Mustang és Corvette

Corvette lesz az AMD mobil Athlon processzorának kódneve. Órajelét még nem közölték. Az év második felében megjelenő chip az asztali gépekbe szánt Mustang párja. E bejelentésen kívül a Wintec konferencián a cég bemutatta a harmadik negyedévre várható AMD-760 chipsetet is, amely már támogatja a DDR memóriát. A DDR legnagyobb előnye, hogy (ellentétben a Rambus-szal) gyártása az SDRAM-éhoz képest csak minimális többletköltséggel jár.

Noteszgépekben 700 MHz

A 700 MHz-es Pentium III notebookok hamarosan megjelennek a piacon. Az Intel SpeedStep energiatakarékos funkciójával felvértezett gépeket elsőként a Dell, a Toshiba és a HP fogja piacra hozni. A Dell Inspiron sorozat tagjai 14,1-15 collos aktívmátrixos kijelzővel, 32 MB memóriával, 4,8–6,4 GB-os merevlemezzel 10–24x CD-ROM meghajtóval, 8 MB-os videokártyával 2300-2800 dollárért lesznek kaphatók. A Toshiba csúcsmodellje 128 MB RAM-mal, 12 GB-tal, 6x-os DVD meghajtóval 4000 dollárba kerül. A HP Omnibook sorozat gépeinek ára 1100-4050 dollár között lesz. (Köztük az új Celeron 550-essel felszerelt gépeké is.) Az asztali gépekbe szánt Celeron 633 és 667 előreláthatólag csak a 700-as változattal egyidejűleg fog megjelenni, hogy biztosíthassák a széles körű terjesztést.

Az Új Celeronok már a Coppermine processzornál alkalmazott 0,18 mikronos technológiával készülnek, tartalmazzák a Pentium III-mal bevezetett SSE utasításkészletet, azonban másodlagos belső gyorsítótáruk továbbra is csak 128 KB, buszsebességük pedig 66 MHz.

Java processzor

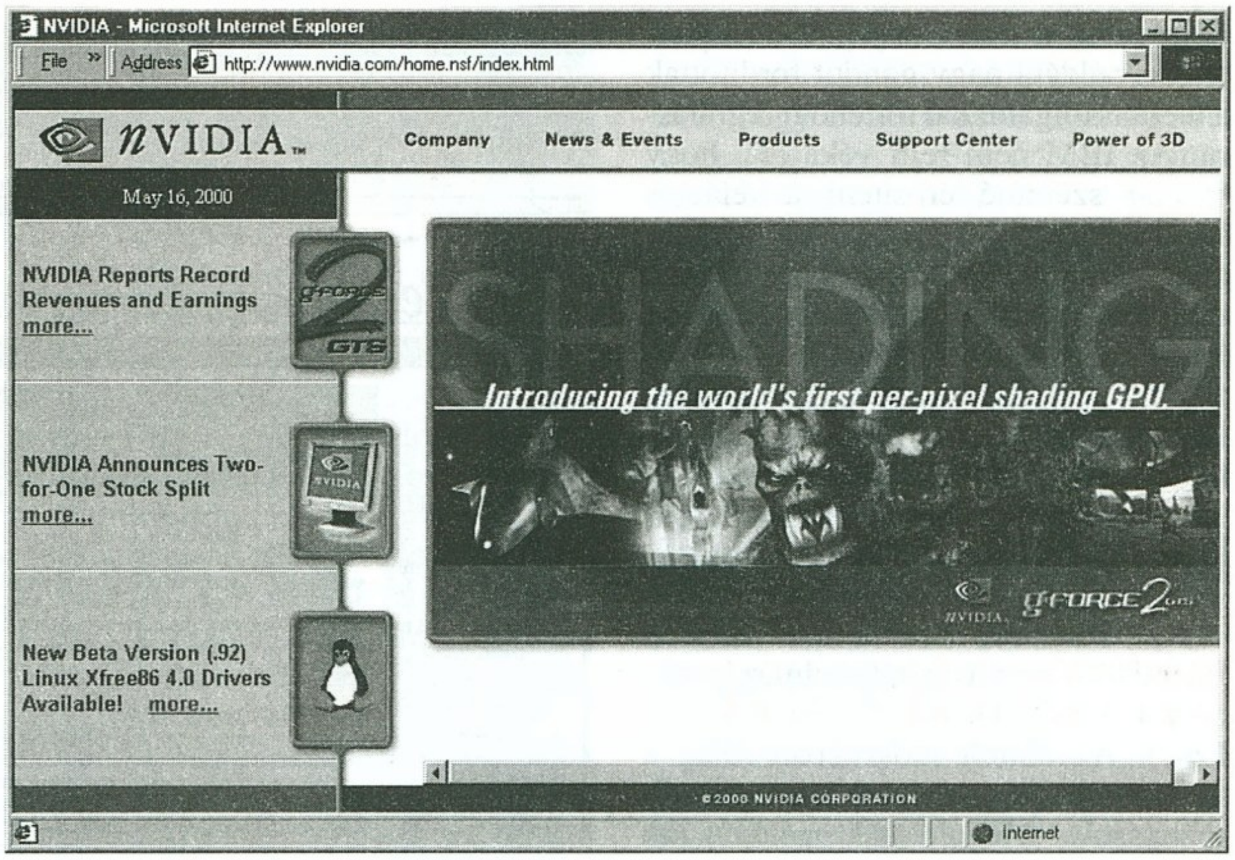
Multimédiát támogató új processzort készít és tesztel a Sun. A MAJC chip (Microprocessor Architecture for Java Computing) multiprocesszor technológiája felgyorsítja a 3D grafikai, video- és hangfelismerő eljárásokat, két 500 MHz-es chip gyorsabb futást tesz majd lehetővé, mint az Intel 1 GHz-ese. A Sun szerint a MAJC chipekből akár 1024 is integrálható lesz.

USB 2.0

Megjelennek az USB 2.0-t támogató rendszerek. A NEC már elkészült az USB host vezérlővel, amely a teljes 480 Mbit/sec adatátvitelt támogatja, és visszamenőleg kompatibilis az USB 1.1 szabvánnyal. A 60 MB/sec új specifikáció 40-szer gyorsabb elődjénél, és ezzel olyan alkalmazások előtt nyit utat, mint a digitális képszerkesztés és az interaktív játék.

Geometriai gyorsító

Az ATI végre ismertette a bemutatókon már felbukkant geometriai gyorsítóját, a Radeon 256-ot. A 0,18 mikronos chip 30 millió tranzisztort tartalmaz, ezzel túlszárnyalja a Pentium III és az Athlon processzorokat, és 128 MB memóriát címezhet a 200 MHz-es DDR RAM-ból. Elvi poligonteljesítménye 30 millió, kitöltési sebessége 1,5 gigapixel (egy képpontra 3 fázist képes egy menetben felvinni, így 500 millió a ténylegesen megjeleníthető képpont), és elsőként tartalmazza integrálva a HDTV dekódert.



A geometriai támogatás a nem látható 3D objektumok vágására is kiterjed (clipping), és a kulcs képkockák közötti átmenet (key-frame interpolation) számításait is elvégezheti, de erre a futtatandó programot is fel kell készíteni. A piacra kerülés időpontjáról és a várható árról még nincs információ. (A geometriai gyorsítót használó nem CAD programokra még várni kell, az ezt kihasználó játékok csak az év végén fognak nagyobb számban megjelenni.) Az ATI Rage 128 Pro chipnél alkalmazott MAXX (kétchipes) megoldásra itt is lehetőség lesz, de csak néhány hónappal később.

A GeForce utóda

Az Nvidia legújabb chipje (korábbi kódneve NV15) a GeForce 2 GTS (Giga Texel Shader) névre hallgat. A processzor órajele 120-ról 200 MHz-re nőtt, a fogyasztás mégis közel a felére csökkent a 0,18 mikronos technológia révén. Több olyan funkció, amely eddig egy sokszögön belül nem volt módosítható (lighting — megvilágítás —, mip-map — távolság függvényében a mintázat változtatása) már pixelenként változhat, ezzel javul a képminőség, és a teljes képernyős antialiasing (élsimítás) is megvalósítható. A maximális poligonszám 25 millió (a GeForce-é 15 millió), a kitöltési sebesség 800 millió kettős textúrázott képpont másodpercenként. 166 MHz-es DDR memóriával működhet, ebből kiskereskedelmi forgalomba a 32 és a 64 MB-os változatok kerülnek, a munkaállomásokhoz pedig 128 MB-os modellek várhatók. A 32 MB-os kártyák induló ára 350 dollár körüli lesz.

Nagyobb felbontású TFT-LCD

Az eddigieknél négyszer nagyobb felbontású TFT-LCD képernyőt fejlesztett ki a mobiltelefonokhoz a Samsung. A kétcollos (kb. 5 cm átlójú) képernyő nagyfelbontású mozgó és állóképek megjelenítésére alkalmas. A reflektív kijelző (szemben a noteszgépekben használt TFT modulokkal) háttérvilágítással, beépített vezérlővel rendelkezik. Tömeggyártása ez év második felében kezdődik.

Nagy kapacitású tároló

Az EMC, a nagyvállalati szerverek tárolóinak gyártója megduplázta a tárkapacitást a Symmetrix sorozatban. A 8000-es tárolórendszer 19,1 terabájt adatot tárolhat, és elődjénél 3-4-szer gyorsabban képes előkeresni azokat. Az adatvédelmet, adatmozgatást, terheléskiegyensúlyozást végző Enginuity vezérlőszoftver a korábbi gépi kódú változatokkal szemben már szinte teljesen C és C++ nyelven készült.

Mobil iroda

A Lotus Notes felhasználói hamarosan mobiltelefonjukról is hozzáférhetnek irodai adataikhoz. A Mobil Notes kézi számítógépekhez és WAP mobiltelefonokhoz lett tervezve. Általa a felhasználók távolról is elérhetik irodai címjegyzéküket, határidőnaplójukat, elektronikus levelezésüket. A Domino szerverekhez az iNotes böngészőn vagy az MS Outlook programon keresztül lehet hozzáférni. A letöltött adatokkal hálózati kapcsolat nélkül dolgozhatnak, a legközelebbi belépéskor mobil készüléken keresztül a cég szerverén is frisstíhetik az adatbázist.

Bánó György
gyuri@mail.vki.bke.hu

A „túlhajszolás” határai

Gyorsítási lehetőségek Abiton

A processzor sebességét illetően telhetetlenek vagyunk. Az órajel növelésével azonban a processzor ára ugrásszerűen nő, és a gyakori cserét csak kevesen engedhetik meg maguknak. Ezért lett népszerű az ún. „overclock”, vagyis a túlpörgetés, a CPU használata a hivatalosan feltüntetett órajelnél nagyobb frekvencián. Az alábbi írás ennek következményeit mutatja be. Példának a hazánkban is — különösen az „overclockosok” körében — népszerű alaplapgyártónak, az Abitnak a modellje, a BE6-II szolgál.

Ma már alig találni olyan hardveres weboldalt, amelyen a processszorok tesztje mellett ne szerepelne a működtetés maximális órajelértéke. Már a 486-osoknál megvolt a túlpörgetés lehetősége, később a Pentiumot pedig részletes cikkek elemezték ilyen szempontból is. Széles körben azonban csak az Intel Celeron sorozatánál terjedt el ez a módszer.

A leglassúbb változatok (Celeron 266 és a 300A) megfelelő alaplapon és az egyik érintkező kiiktatásával 66 MHz helyett 100 MHz-es külső órajelen tudtak működni, elviselve az ezzel járó 50%-os gyorsítást. (Ennek a lehetőségnek az a magyarázata, hogy az azonos kategóriájú processzorok ugyanazon a gyártósoron készültek, és csak a cég üzletpolitikája határozta meg, hogy hány darab lett 300, és hány 500 MHz-es.)

Mivel az Intel a processzorok szorzóját (300-as esetén 4,5x66 MHz) újabb termékeiben már rögzíti, a magasabb órajelű változatoknál ilyen eredmény elérésére csekély az esély, ráadásul a Intel BX chipsetek hivatalos sebessége is csak 66 vagy 100 MHz lehet, amit a legtöbb gyártó be is tart, mert az ettől eltérő értékek az alaplapra csatlakozó vezérlőkártyák órajelét is a nem szabványos sebességre állítják, míg 100 MHz-nél csak a processzortól függ a túlhajtott rendszer „életképessége”.

A mellékelt PCI/AGP táblázatból látszik, hogy szabványon kívüli esetekben hogyan változnak az órajelek. A vastaggal szedett értékeket a kártyák többsége még tolerálja, de az ezt meghaladó órajelet elviselő eszközökhöz már csak szerencse vagy próbálgatás útján lehet hozzájutni. (Igazán stabil túlhajtott gépet építeni nem művészet ugyan, de türelemjáték, és a kockázat saját zsebre megy, a garancia erre nem vonatkozik!)

Elsőként az Abit

Az Abit volt az első olyan gyártó, amely az alaplap órajelének változtatását szoftveresen, a BIOS-on keresztül oldotta meg meg, leegyszerűsítve a folyamatot, és megszabadítva az önjelölt tuningolókat a gépben való turkálástól. Jelenleg a Softmenu BIOS a harmadik változatánál tart, az órajel 83 MHz-től 200 MHz-ig egyesével léptethető. Ilyen található a BF6 és a BE6-II alaplapon is. Mindkét alaplap Intel SLOT1 processzorokat fogad, Celerontól a Coppermine-ig. A BE6-II plusz szolgáltatása a saját BIOS-szal rendelkező kétportos UDMA-66 vezérlő, ami az alapkiépítés 4 EIDE meghajtója mellé további 4 csatolását teszi lehetővé.

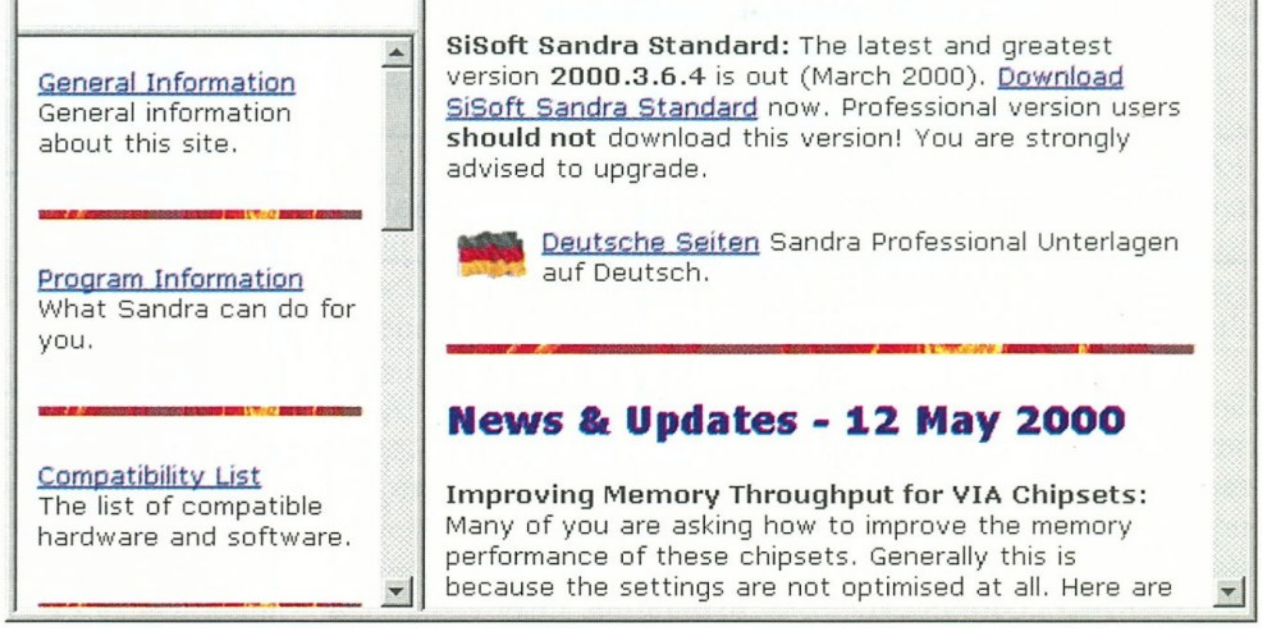
A túlhajtáshoz 100 MHz-ig mindenki könnyen eljuthat, ha veszi magának a bátorságot és a megfelelő Celeront, de e felett már sokkal nehezebb alkalmas alanyt találni. A tesztben egy gyári hűtőbordával és ventillátorral ellátott Intel 350-es Pentium II processzoron próbáltam a lehetséges maximumot megtalálni.

Ez a processzor már egyáltalán nem mondható fiatalnak, az első volt a 100 MHz-esek között, szinte egyidejűleg jelent meg a 400/450-esekkel, és megvolt az esély a sikeres felpörgetéshez. Az alaplaphoz külön adtak egy kis füzetet, amely BX 133 Overclocking Guide névre hallgat. Az abban felsorolt 46 grafikus kártya elviselte a 133-as órajelet, valamint az ajánlott PCI és AGP osztókat.

Elég sok órajelen kipróbáltam, közülük 4 fázist érdemes kiemelni:

1. Tesztek futtatása hivatalos gyári értéken

Itt megemlítem a gépen lévő főbb alkatrészeket is: Abit Siluro TNT2 Ultra AGP 32 MB-os képernyővezérlő (150/183 MHz), Diamond S90 PCI hangkártya, 2x64 MB-os NPC 8 ns SDRAM, Quantum FB 6,4 GB-os merevlemez az UDMA66 vezérlőn, és Fujitsu 3,5 GB-os az UDMA33 ágon. Operációs rendszer a Windows 98, DirectX 7a-val. A videokártya-gyártásba az Abit csak nemrég kapcsolódott be,



a TNT2 Ultra az NVidia referenciamodelljére épül, az analóg VGA mellett, s-video és digitális monitorkimenetet tartalmaz. (Sajnos az s-video kimenet csak a VGA-val felváltva működhet, ami nem a legkényelmesebb megoldás.)

2. Futtatás 120 MHz-en

Minden BIOS érték maradt a 100 MHz-es beállításain, a gép teljesen stabil maradt, semmilyen probléma nem volt észlelhető.

3. Futtatás 140 MHz-en

Ez már nagy ugrás, fontos beállításokat kellett módosítani a BIOS-ban. Az első a memória elérési ciklusának (CAS) beállítása 2-ről 3-ra (kettes módban a fenti memória 128 MHz-ig működött), a PCI osztó 1/3-ról 1/4-re állítása. A gép így rendesen elindult, de a 3D programok szinte azonnal lefagytak, ezért az AGP módot is 1x-re kellett váltani (ez a Softmenu-ben a Fast-ról Normal-ra való átállítást jelenti). Bár az AGP átviteli teljesítmény elvben felére esett vissza, gyakorlatban a memória sebességének csökkenése jobban befolyásolta az eredményeket.

Mivel a tesztek csak az első merevlemezt használták, később vettem észre, hogy a második meghajtón írási hiba keletkezett (FAT) a túl magas PCI órajel miatt, az alaplap és a winchester vezérlője nem tudott megfelelően kommunikálni, a scandisk pedig nem mindent tudott helyreállítani. Ezt a BIOS-ban az erre a merevlemezre vonatkozó opció UDMA-ról a lassabb MODE-3-ra állításával orvosoltam. Ez a leggyakrabban felmerülő problémák egyike. Célszerű elmenteni fontos adatainkat, ha overclockra adjuk a fejünket!

4. Futtatás 146 MHz-en

Itt a processzor már csak úgy volt hajlandó működni, hogy feszültségét 2 V-ról 2,1-re növeltem. Mivel a 140 MHz-nél már minden lehetséges osztó a minimumra került, nem is lett volna lehetőség további változtatásokra. Egyes tesztprogramok ugyan lefutottak még 150 MHz-en is, de a rendszer rövid időn belül minden 3D programtól leállt, vagy megszakította azok futtatását.

A teszteredményekről

A táblázatban a megszokottól eltérően most csak az első hasáb tartalmazza a tényleges értékeket, a többi a százalékos változást jelzi a gyári értéken mérthez képest. A CPU, az FPU és memória sebességének mérésére a SiSoft Sandra 2000 standard (http://www.sisoftware.co.uk/sandra), a CAD OpenGL alkalmazásokhoz a már korábban is használt Indy3D (www.sense8.com/indy3d), a játékteszthez pedig a Quake 3 Arena program első demóját futtattam.

A processzor nyers teljesítménye — ahogy az várható volt — teljesen lineárisan változott az órajellel. A memória is hasonlóképpen, egészen addig a pontig, amíg nem kellett a beállításait megváltoztatni. A játéktesztben is szépen nő a teljesítmény az 1024-es felbontásig, ahol viszont már a videokártya képessége korlátozza a processzort.

A tesztben használt videokártya túlhajtható 10%-kal, de ez csak az 1024-es és annál nagyobb képméretnél hoz változást (5-9%), mert a processzor lassabb annál, hogy ez alatt elérje a kártya határértékeit. A CAD rendszereket a nagy lebegőpontos számításigény miatt kisebb mértékben befolyásolta a memória sebességének csökkenése, mint a Quake 3-at, mégis a CAD munkakörnyezet viseli el inkább a drágább kategóriájú processzorok megvásárlását, mert a tervezőrendszerek ára még így is nagyságrendekkel meghaladhatja a „vas" árát.

Ha a processzorok között továbbra is lesznek túlhajtásra alkalmas példányok, valamint akadnak alaplapgyártók, amelyek támogatják az overclockot, akkor e módszer népszerűsége tovább nőhet a szűkebb pénztárcájú otthoni felhasználók körében. A Coppermine és a Celeron processzorok új SSE utasításkészletű szériái is hasonló lehetőséget rejtenek, de korlátozott számuk és magas áruk miatt ezek hazánkban egyelőre kevésbé elérhetőek.

Friss információk

Az alaplap kiválasztása a gépösszerakás egyik legkritikusabb pontja, még akkor is, ha nem akarjuk rendszerünk összhangját túlhajtással megbontani. Válasszunk olyan alaplapot, amely egy kategóriával gyorsabb processzort is képes fogadni, így a következő bővítést féláron megúszhatjuk.

Az Intel BX chipsetje lassan már kétéves, mégis tartja vezető pozícióját, mert a újabb chipkészleteket a RAMBUS memóriához tervezték, amelynek ára 6–8-szorosa a SDRAM-nak, teljesítményét pedig még nem is sikerült kihasználni. A márkás alaplapok jelenleg 5–10 ezer forinttal drágábbak a „no name" gyártmányoknál, de szolgáltatásban, frissítésben és stabilitásban felülmúlják azokat.

Az internet tele van tesztekkel, melyekhez olvasói fórumok is kapcsolódnak, ezért óriási előnyhöz jut az, akinek van internet-elérése. A szűkös hazai információforrásokat a világháló ugrásszerűen kibővíti.

Bánó György

PCI/AGP órajelek alakulása az alaplap órajelének függvényében

(1/4-es PCI osztó a BX-es lapokon általában nincs)

Alaplap	PCI 1/2	PCI 1/3	PCI 1/4 (BE 6-II)	AGP 1/1	AGP 2/3
66	33	22	17	66	44
75	38	25	19	75	50
83	42	28	21	83	55
100	50	33	25	100	67
120	60	40	30	120	80
133	67	44	33	133	89
140	70	47	35	140	93
146	73	49	37	146	97

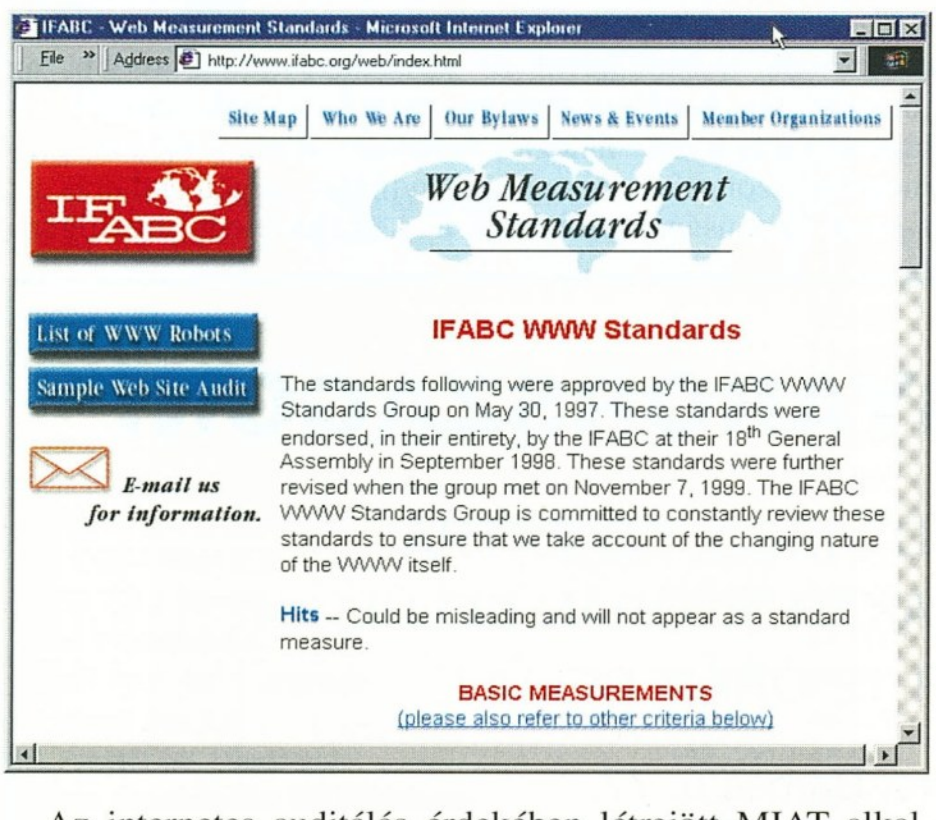
Órajel	350	420	490	510
Órajel növekedése		20%	40%	46%
SiSoft Sandra 2000				
CPU	945	20%	40%	46%
FPU	469	20%	40%	46%
CPU Multimedia	1080	20%	40%	46%
FPU Multimedia	504	20%	40%	46%
Memória CPU	308	20%	21%	27%
Memória FPU	312	20%	29%	35%
Indy3D				
Mcad40	7,15	20%	36%	42%
Mcad150	2,2	20%	35%	42%
Animation	8,02	20%	36%	42%
Simulation	34,4	20%	35%	42%
Quake 3				
1024, 16 bit	33,8	20%	22%	25%
1024, 32 bit	31,8	13%	16%	16%
800, 32 bit	33,8	20%	29%	34%
640, 32 bit	34,3	20%	31%	36%

Indul az internet-audit

Közel egy évnyi munkával elkészült az „Internet Audit Szabvány", és azt 2000. május 15-én az érdekelt szervezetek elfogadták. A dokumentum meghatározza az internetes oldalak látogatottságára vonatkozó fogalmak definícióit és az auditálás (hiteles ellenőrzés) eljárását, melyek alapján Magyarországon is megkezdődhet az egységes internet-audit szolgáltatás. A MIAT (Magyar Internet Audit Tanács) azzal a céllal jött létre, hogy koordinálja a fenti szabvány elkészítését. A szervezetnek tagja a DME (Direkt Marketing Egyesület), az IVSZ (Informatikai Vállalkozások Szövetsége), a MAK (Magyar Adatbázisforgalmazók Szövetsége), a MARS (Magyarországi Reklámügynökségek Szövetsége), a MATESZ (Magyar Terjesztésellenőrző Szövetség) és az MRSZ (Magyar Reklámszövetség).

A nyomtatott sajtótermékekre vonatkozóan már hosszabb ideje (a fejlettebb országokban több évtizede) működik a példányszámadatok auditálási rendszere. Ennek lényege, hogy a nyomtatott példányszám és a terjesztés adatainak ellenőrzése és hitelesítése céljából a kiadóknál erre a témakörre leszűkített könyvvizsgálói auditálást végeznek, meghatározott rendszerességgel. A közölt adatok hitelességét az auditálást lebonyolító független szervezet (Magyarországon a MATESZ) és a megbízott hiteles könyvvizsgáló együttesen garantálja. Az auditáltatást önkéntesen vállalók tulajdonképpen arra kötelezik magukat, hogy hirdetésszervezőik nem fognak légből kapott példányszámokkal és terjesztési adatokkal, tehát tisztességtelen módszerekkel manipulálni a hirdetők meggyőzése érdekében.

Bár „az olvasottságot" az interneten is kezdettől fogva mérik, az adatok hitelesítése a sajátos technikai feltételek miatt elég lassan alakult ki. Szinte minden online média közölt magáról látogatottsági adatokat, de ezek nem mindig voltak összehasonlíthatók, még akkor sem, ha a működtető kiadókat semmiféle megtévesztési szándék nem vezérelte. A hitelesítés megoldatlan módszertani problémái sokáig nehezítették az auditálás egységes tartalmúvá és általánosan elfogadottá válását.



Az internetes auditálás érdekében létrejött MIAT alkalmazza a Nyugat-Európában már elfogadott IFABC szabványt. (International Federation of Audit Bureaux of Circulations, http://www.ifabc.org, a MATESZ 1996 óta tagja ennek a szervezetnek.) A MATESZ által használt rendszer a szoftvereket, adatrögzítési és ellenőrzési technikákat is beleértve teljesen átlátható. A mérési adatok hamarosan elérhetők lesznek a www.matesz.hu weblapon, a szabvány pedig megtekinthető lesz a www.miat.org honlapon. A MIAT szeretné elérni, hogy az internetes reklám elhelyezése során a hirdetők ezekre a hiteles látogatottsági adatokra támaszkodjanak, és a weblapok saját magukról közölt adatai se térjenek el azoktól. Ezáltal összességében is nőhet a bizalom az internetes reklámhordozók iránt.

A virtuális áruház magja

Az elektronikus kereskedelem területén próbálkozó cégek egy része nem rendelkezik megfelelő programfejlesztő háttérrel, és azt nem is akarják kialakítani mindaddig, amíg az üzlet igazán be nem indul. Eltekintve attól, hogy ez egyfajta ördögi kör, nem is kell feltétlenül az alapoktól építkezni. Forgalomba kerültek ugyanis a „csinálj magad e-áruházat" jellegű fejlesztőeszközök, és megjelent a virtuális üzletbérlet. Ezekben hasznosíani lehet azokat a módszereket, amelyekkel egy-egy nagyobb cég felépíti több virtuális osztállyal felszerelt áruházát.

A virtuális áruházak megvalósítását szolgáló szoftverek közül a Store Wizard (Laurel Kft) magyar fejlesztésű. Első



verziója 1998-ban került forgalomba, és jelenleg három változatban készül. A Micro Store maximum 50 alaptermék forgalmazására, paramétereinek tárolására van felkészítve, a Medium Store 500, a Mega Store korlátlan számú termék virtuális készletezését teszi lehetővé. Közös jellemzőjük a szerveroldali szoftvermegoldás, a kliensgépeken pedig böngészős környezetben jelenítik meg az adminisztrátori és bolti kezelőfelületet. Azoknak a cégeknek, melyek nem akarnak saját szervert működtetni, lehetőségük van a forgalmazó Store Wizard szerverén boltot bérelni, megtartva az egyedi megjelenés lehetőségét, minden egyes üzlethez saját arculat (design) terveztetésével.

A hazai fejlesztésű termékek nagy kérdése, hogy meg lehet-e jelenni velük a nemzetközi piacon is. A Store Wizard ebben a tekintetben elég jól áll, mert a fejlesztés az e-business üzletágban jóval fejlettebb amerikai piacra irányult, és internetes óraboltjuk (http://www.worldofwatches.com) több elemző szerint műfajában az egyik legjobb. Ugyanezzel a

szoftverrel működik a hazai internetes piacon a http://www.libri.hu és a http://www.swsbooks.hu webüzlet, a Fotex internetes áruháza, valamint a MatávNet aukciós rendszere. A Store Wizardról további információk a http://www.laurel.hu oldalról szerezhetők be.

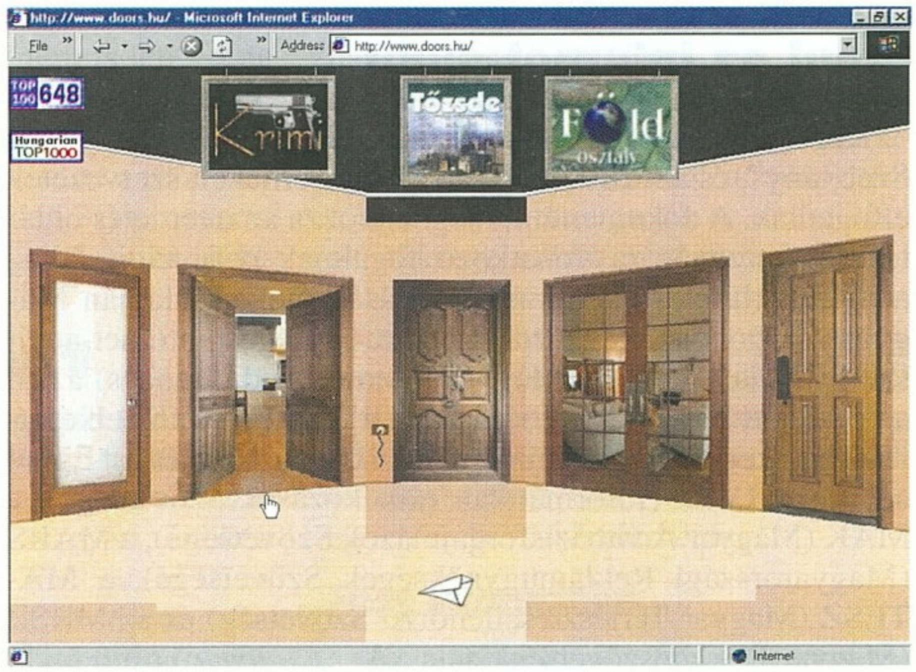
Norton Antivirus Linuxra

Linux platformon eddig nem sok vírus csinált karriert, de az internet biztonságának kérdéséhez hozzátartozik, hogy ezt a platformot sem lehet védelem nélküli hagyni. Különösen, hogy az internetes kiszolgáló szervergépek jelentős része Linux operációs rendszerrel fut, de más célokra szolgáló szervereken is egyre többen választják a Linuxot. A vírusfejlesztőknek nagy kihívás lehet a Linux ELF-jeinek megtámadása, bár a makróvírusok és a VBScript-vírusok sikeréről a Linux kórokozói nem is álmodhatnak. A Symantec május elején bejelentette, hogy a Norton Antivirus programot elkészítette Linux platformra is. Ebben szintén Navex motor dolgozik, és online kapcsolatban ehhez is rendelkezésre áll a LiveUpdate szolgáltatás.

Kezdje az előszobában...

Újabb „széles spektrumú" felhasználói kiindulóhely lépett színre: a Doors (ajtók) elnevezésű magyar nyelvű webportál. Arculata, grafikai és navigációs megoldása eltér a meglévőkétől. Tervezői az egyszerűségre és a jó áttekinthetőségre törekedtek, a látogatók egy virtuális lakás előszobájából nyíló ajtókon keresztül indulhatnak útnak. Azokat az információkat helyezték el a webhelyen, amelyek az otthoni használathoz nagy valószínűséggel hozzátartoznak. A www.doors.hu középpontjában ugyanis a család áll. A napi családi élet, a beszélgetések, a közös feladatok megoldása közben felmerülő megannyi téma, kérdés, terv... A portál szerkezeti felépítése igyekszik figyelembe venni, hogy a család tagjainak érdeklődése, és ennek megfelelően barangolási iránya is eltérő. A tematikát jól mutatja a választható pontok listája:

Technika — Család — Utazás — Életmód — Vásárlás — Internet — Otthon — Irodák — Sport — Könyv, zene — Mobil — Konyha — Klubok — Egészség — Jegyek — Audio, Video — Családi vagyon — Célpontok — Gasztronómia — Háztartás — Freeweb — Esküvő — Útleírások — Tudomány — Antikvitás — E-mail — Ünnepnapok — Szálláshelyek — Könyvtár — Szoftver — Számítógép — Hobbi — Kereskedelem — Oktatás — Hardver — GPS — Játék — Kaland — Szórakozás — Ruházat.



A portálhoz csatlakozott (nem csak linkkel rámutatott) partnerek információit folyamatosan frissítik. A megjelenés további csiszolása során pedig a portál nyitóoldalát jelentő előszoba mindig az adott napszaknak megfelelő életképpel fog bejelentkezni.

Elektronikus biztonság

Az utóbbi egy évben sok gondot okoztak a szoftverek biztonsági hiányosságai. Az is nyilvánvalóvá vált, hogy az elektronikus kereskedelemnek — és általában az internet használatának — komoly akadálya lehet, ha nem sikerül kidolgozni és elterjeszteni a kellőképpen biztonságos technológiákat, másrészt ha az emberek bizalmatlanok maradnak az elektronikus ügyintézés iránt. Az ipar gyorsan reagált erre a kihívásra, például azzal, hogy szakmai szemináriumok, konferenciák, kiállítások egész sorát szentelték ennek a támakörnek. Magyarországon még nyár elején is szinte egymásba érnek az ilyen rendezvények. Egy részük borsos részvételi díjat, más részük csak jelképes összeget kér (ami ma már persze általában 10 ezer forint), de vannak ingyenesek is, amilyen a június 20-22. közötti ikerrendezvény: a Cards és az Information Security kiállítás és konferencia a Budapesti Kongresszusi Központban. Az előbbi az intelligens chipek és kártyák technikájával és használatával, az utóbbi az információtechnológia védelmi megoldásainak egész palettájával foglalkozik: hálózati behatolás elleni védelem, biometrikus azonosítás, biztonsági auditálás, digitális aláírás, az e-mail biztonsága, elektronikus kereskedelem, intelligens

kártyák, kriptográfia, operációs rendszerek védelme, titkosítási eljárások, távelérés, tűzfalak, virtuális magánhálózatok, vírusvédelem. Természetesen az Új Alaplap sem marad ki a biztonságtechnikai ismeretterjesztésből: augusztusi számunkban a hónap témája szintén ezzel foglalkozik. (Aki azt gondolná, hogy a frissiben támadt konjunktúrahullámon lovagolunk, eláruljuk hogy mi ezt az augusztusi témát már egy évvel korábban beütemeztük.)

Genius márkaképviselet

A Genius eszközöket gyártó tajvani Kye Systems fejlesztőgárdája az adatbeviteli eszközök kialakítását ma is meghatározó számos szabadalmat mondhat magáénak. Például 1998-ban az internet használatát megkönnyítő görgető funkciót (scrolling) a Genius NetMouse egércsaládjánál alkalmazták először. (Ezt a megoldást a Microsoft is „átvette", ami maiatt a Kye pert indított, de egy peren kívül keresztlicenc megállapodás eredményeként a két cég ezen a területen ma már egyezteti fejlesztéseit.)

Piacelemző cégek szerint a világon tavaly eladott 100 millió PC közül minden hetedikhez Genius márkajelzésű egeret használnak, emellett a számítógépgyártók saját márkafeliratú egereket is rendelnek tőlük. Új Genius termék a nagyfelbontású, diaképek beolvasására is alkalmas szkennercsalád, az új technológiát alkalmazó optikai egér és az energiatakarékos „Green" hálózati eszközpark. Az internetes adatletöltést csavart érpáros hálózaton meggyorsító ADSL technológia is saját fejlesztésük ereménye. A letöltés sebességet növelő PCI csatlakozású termékek bejelentését követően idén megkezdődik az ADSL routerek gyártása is. A Genius márkaképviseletet Magyarországon 2000. április 25-től kizárólagos disztribútori joggal a Fan Electronics (www.genius.hu) látja el, az előzőnél jóval nagyobb alapterületű, sokoldalú szolgáltatást lehetővé tevő új telephelyén.



A Dell és a VMware

A VMware iránti érdeklődésünkkel (és a 10. oldalon olvasható cikkünkkel) nagyon jól összecseng az a lapzártakor érkezett a hír, hogy a Dell Computer 20 millió dollárt fektetett a VMware szoftvert kifejlesztő cégbe. A VMware 30 napos próbaváltozata CD-mellékletünkön is megtalálható, Windows NT/2000, illetve Linux operációs rendszerre. A telepítés után kapott virtuális gép a rendszerek „békés egymás mellett élésének" nagy reményekkel kecsegtető eszköze. A Dell mostani lépése és a többi befektető megjelenése (Azure Capital Partners, Chase H&Q Capital Partners, Goldman Sachs Group) vélhetően felgyorsítja a VMware fejlesztését és a számítástechnika „belső kompatibilitását" segítő más alkalmazások létrehozását.

Rendszer és rendszeretet

Betömni az oprendszerek hézagait

Egy számítógép működtetése során rengeteg olyan probléma merülhet fel, melyek megoldására — vagy egyáltalán a hiba okának megtalálására — az operációs rendszer még oly kiforrott eszközei sem elegendőek. Ilyenkor lépnek színre azok a hasznos segédeszközök, melyek befoltozzák az operációs rendszerek egyes hiányosságait. A Közkincs anyagának mostani válogatásakor a System Internals termékei között „egy bokorban" több ilyet is találtam.

Regmon

A Regmon feladata a registry folyamatos figyelése és az összes olyan esemény feljegyzése, melynek során egy program lekér a registryből valamit, vagy módosítást hajt végre benne. Felhasználható a rendszerkonfiguráció (registry bejegyzések) hibáinak kiderítésére nemcsak a működő rendszerben, hanem már a betöltés során is. Ellenőrizni lehet vele a telepítések menetét vagy a programok működését, hogy kiderüljön róluk, milyen rejtett bejegyzéseket használnak. El lehet kapni vele a programok registryhez fordulását, és szűrni lehet azt. Kezelése kényelmes, egyszerű. Telepítésre nincs is szükség, a programot csak be kell tenni egy könyvtárba, és onnan bármikor indítható. Sokszor tehet jó szolgálatot. Mérete mindössze 158 KB. Forráskódban is hozzáférhető (lásd a CD-mellékleten).

Név: Regmon
Típus: Registry Monitor
Verzió: 4.24
Platform: 9x/NT
Fejlesztő: System Internals
Honlap: http://www.sysinternals.com
Licenc: Ingyenes

Portmon

A Portmon az előző program testvére. Feladata a soros és a párhuzamos portokon keresztülmenő forgalom figyelése (monitorozása). Felépítésében és használatában is nagyon hasonlít a Regmonhoz, a soros vagy párhuzamos portok használata során felmerülő problémák kiküszöbölésére érdemes használni.

Név: Portmon
Típus: Portfigyelő
Verzió: 3.01
Platform: 9x/NT
Fejlesztő: System Internals
Honlap: http://www.sysinternals.com
Licenc: Ingyenes

Filemon

Nem csupán azt tudhatjuk meg ezzel a monitorozóval, hogy melyek az éppen megnyitott fájlok, hanem azt is, hogy az egyes állományokat melyik program nyitotta meg. Elsősorban annak kiderítésében tehet jó szolgálatot, hogy egy állományt miért nem tudunk módosítani. Azért-e, mert valamelyik másik program jogosan megnyitotta, vagy mert egy illetéktelen program — például vírus — szerkesztgeti azt.

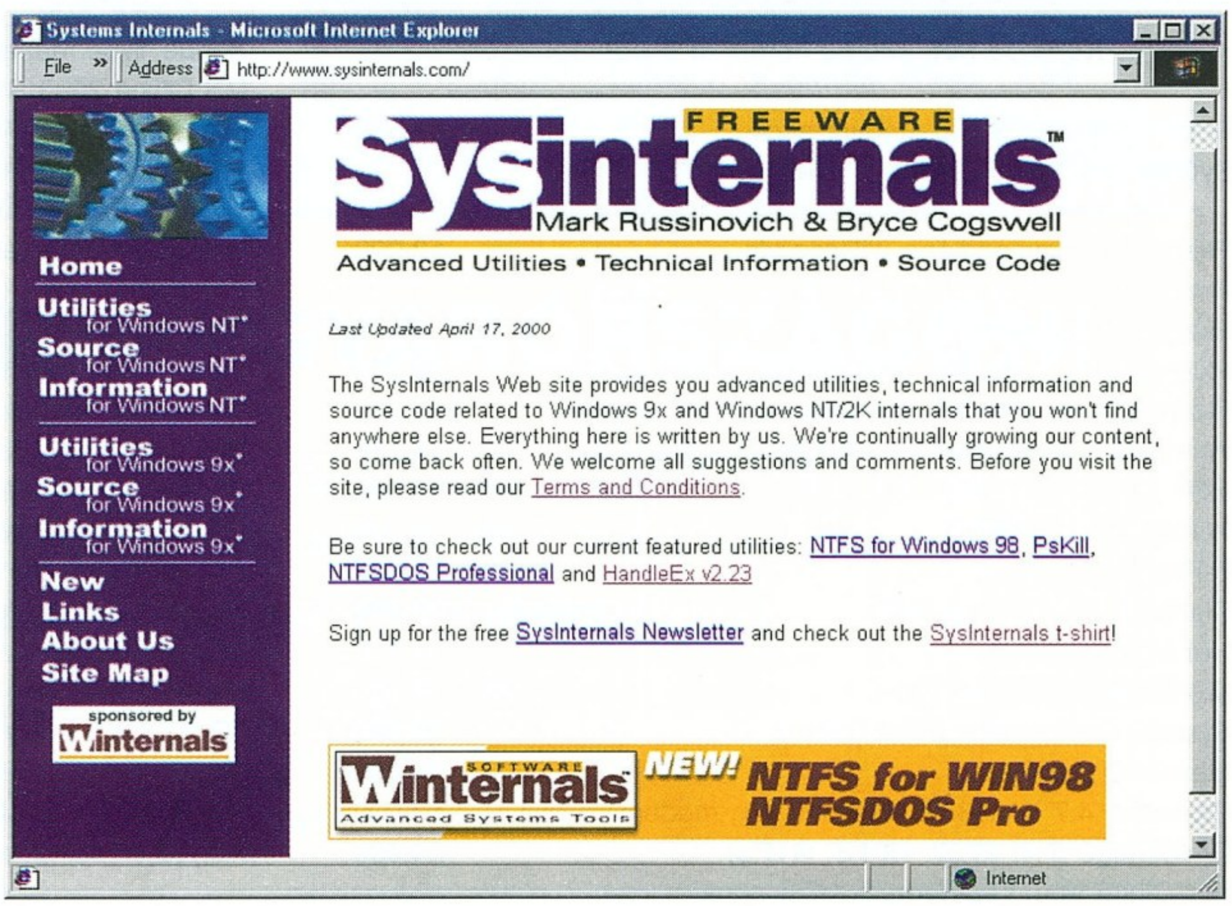
Név: Filemon
Típus: File Monitor
Verzió: 4.28
Platform: 9x/NT
Fejlesztő: System Internals
Honlap: http://www.sysinternals.com
Licenc: Ingyenes

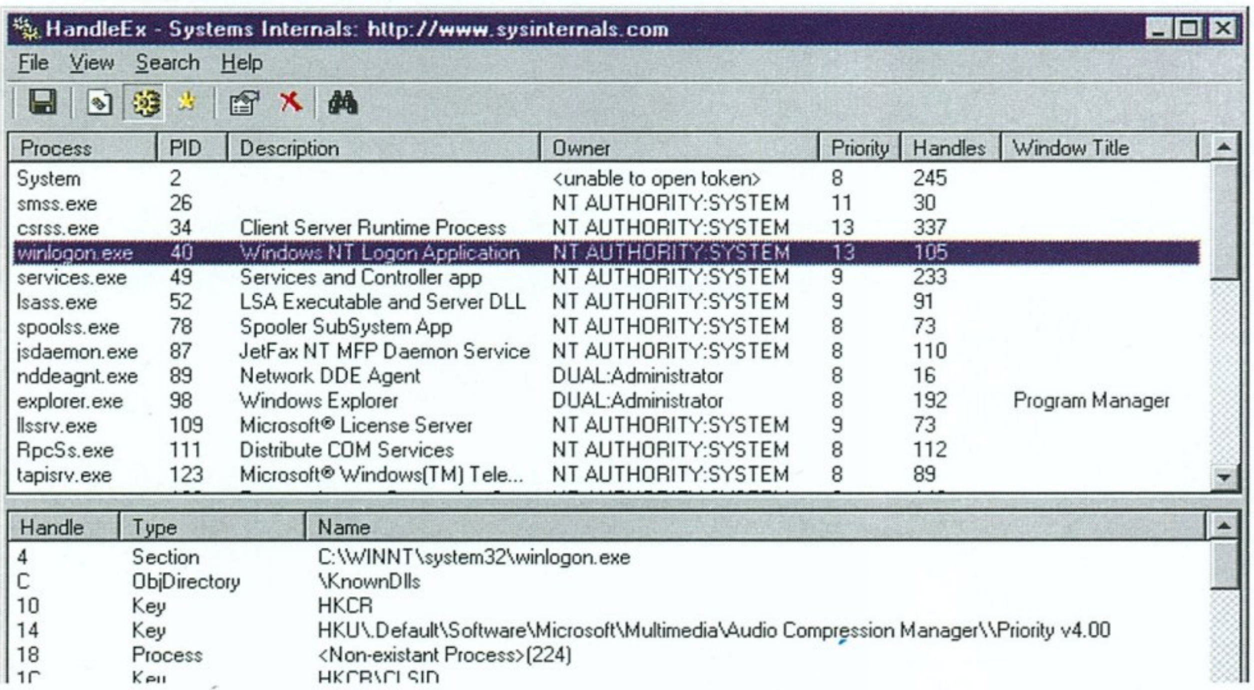
HandleEx

Az éppen futó alkalmazásokat, illetve az általuk lefoglalt erőforrásokat tekinthetjük meg ezzel a programmal, ide értve a megnyitott fájlokat, az elindított szálakat, a használatban lévő portokat és szemaforokat, a betöltött DLL-eket stb.

Kiváló eszköz a programozási hibák felderítésére, ha ugyanis valaki lefoglal erőforrásokat, majd nem szabadítja fel azokat, ez a program jelzi a hibát. További hasznos szolgáltatása, hogy segítségével az egyes folyamatokat (process) ki lehet kapcsolni (kill), tehát úgy működik, mint egy továbbfejlesztett Task Manager.

Név: HandleEx
Típus: Erőforrás kijelző
Verzió: 2.23
Platform: NT
Fejlesztő: System Internals
Honlap: http://www.sysinternals.com
Licenc: Ingyenes





Contig

A számítógépen tárolt állományoknak a merevlemezen való elhelyezkedése előbb-utóbb töredezetté válik, ami csökkenti a teljesítményt. Írtak már számos kiváló programot, melyekkel a lemezen ezt a szétszóródást meg lehet szüntetni. Csakhogy egyre nagyobbak a merevlemezek, azokon egyre több adatot tárolunk, ezért egyre hosszabb időbe telik az összetartozó adatok összefüggő blokkokba rendezése. Ha ritkán végezzük el ezt a korrigálási feladatot, akkor lelassul a rendszer, viszont túl gyakran azért nem lehet sort keríteni rá, mert a művelet nagyon hosszadalmas. A megoldást a Contig program kínálja. Nem kell az egész lemezt, annak minden állományát felkutatni és átrendezni, mert módunkban áll külön egyes fájlok töredezettségét is megszüntetni. Vagy lehet például a fontos rendszerállományok rendszeres karbantartására koncentrálni, hogy a komplett, átfogó rendezésre minél ritkábban legyen szükség. Használható a program a fájlok töredezettségének ellenőrzésére is, így villámgyorsan megtudhatjuk és eldönthetjük, hogy érdemes-e a töredezettség teljes megszüntetését végigcsinálni.

Név: Contig
Típus: Töredezettség-mentesítő
Verzió: 1.3
Platform: NT
Fejlesztő: System Internals
Honlap: http://www.sysinternals.com
Licenc: Ingyenes

ERD Commander

Az ERD Commander lehetővé teszi a Windows NT rendszer betöltését floppyról, ezáltal kideríthető a rendszerleállások oka, és kijavítható a hiba. A floppyk elkészítéséhez szükség van az NT telepítőlemezeire is. (Azt ugyanis nem lehet beállítani, hogy a telepítőkészletet a hálózaton vagy egy helyi könyvtárban keresse, mindenképp a CD-hez akar fordulni.) Ha a floppyk nem állnak rendelkezésre, akkor az ERD Commander először elkészíti azokat. Ha rendelkezésre állnak (vagy elkészültek), akkor módosítja azokat úgy, hogy betöltés után a hagyományos telepítés vagy helyreállítás választási lehetősége helyett a rendszer elinduljon a parancssorban. Ez a program sajnos nem teljesen ingyenes, de bizonyos korlátozásokkal szabadon használható. (A korlátozás az, hogy a fájlokat csak olvasható módon kezeli.) További hiányossága, hogy ha a felhasználó olyan partíciót is használ, amelynek fájlrendszere csak kiegészítőkkel érhető el (például FAT32-vel), akkor azt a partíciót egy ilyen betöltés után nem lehet elérni.

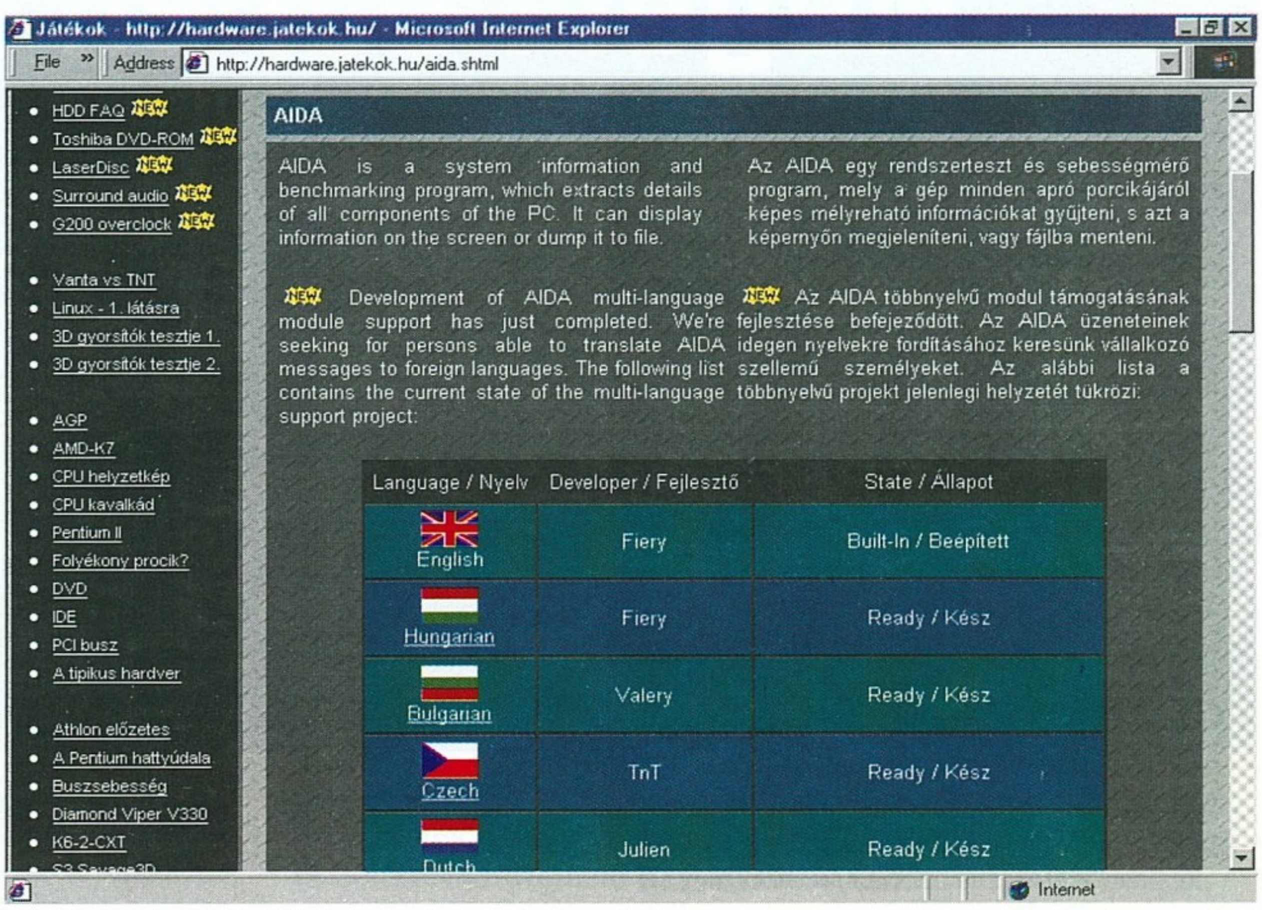
Név: ERD Commander
Típus: NT betöltése floppyról
Platform: NT
Fejlesztő: System Internals
Honlap: http://www.sysinternals.com
Licenc: Ingyenes (read only)

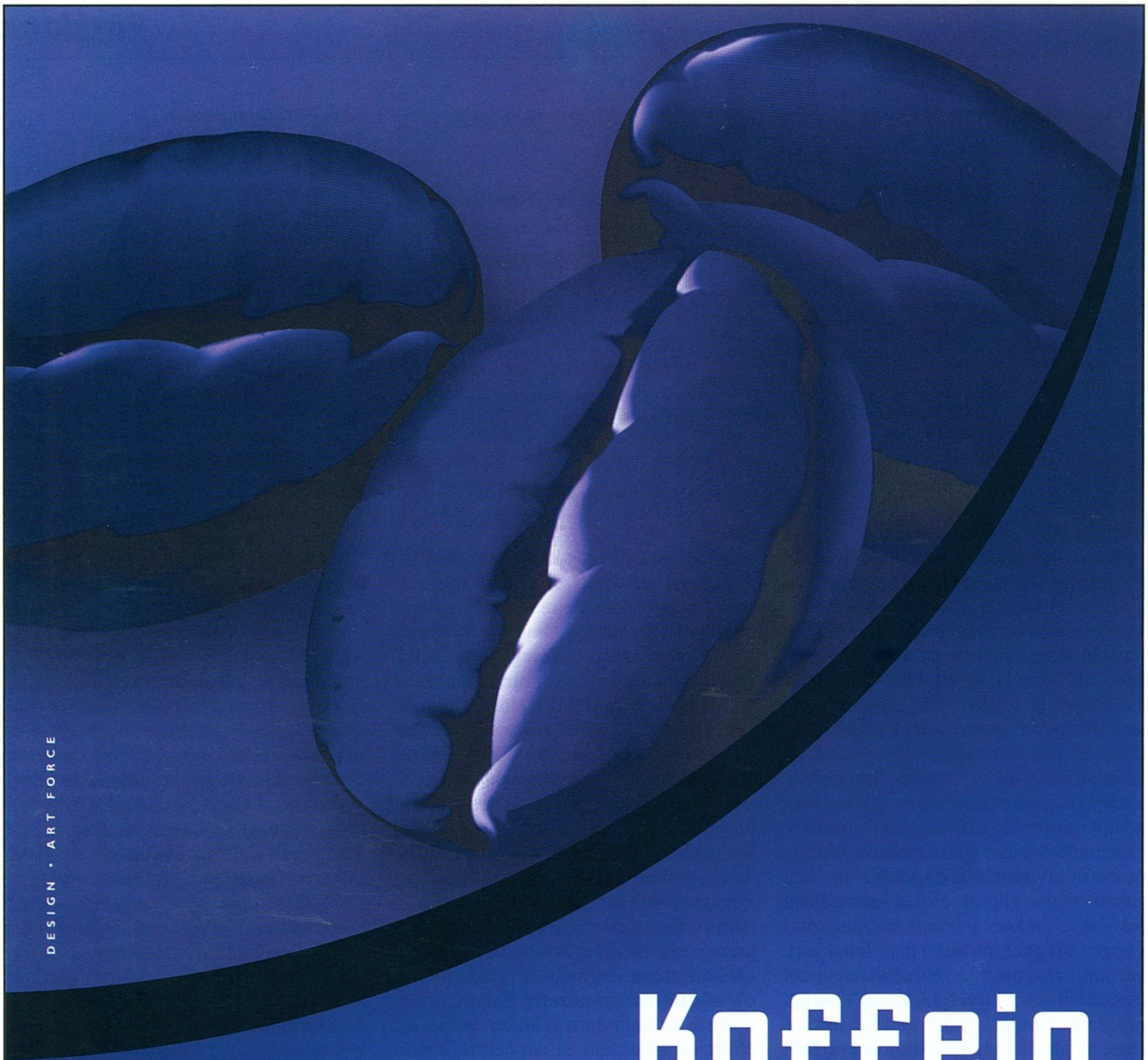
Aida

Ez a program az egyik legteljesebb, legnagyobb lehetőségekkel rendelkező számítógéptesztelő program (hardverre és szoftverre egyaránt). A jól ismert ASM Demo utódja, és — ami szintén nem elhanyagolható — magyar fejlesztő terméke. Naprakész, pontos és kimerítő információkat ad a számítógép minden porcikájáról, a processzortól a BIOS-on és a chipkészleten át a perifériákig. Támogatja és felismeri az összes létező PC-s processzortípust, ismeri azok minden paraméterét, ismer több mint kétezer alaplap típust, számos hangkártya, videokártya és hálózati kártya típust és egyéb perifériát. Használata jó szolgálatot tehet új hardverelemek beszerzésekor (kevésbé csaphatják be az embert), vagy új gép összeszerelésekor (megfelelő-e minden eszköz), illetve eltérő konfigurációk vagy hardverelemek összehasonlításakor. Csak otthoni használatra ingyenes!

Név: Aida
Típus: Tesztelő
Verzió: 1.00-0511
Platform: 9x/NT/DOS
Fejlesztő: Miklós Tamás
Honlap: http://hardware.jatekok.hu/aida/
Licenc: Ingyenes (otthoni használatra)

Nagy Tamás
bigtom@avalon.aut.bme.hu




DESIGN · ART FORCE

Az IPv6 mobil próbája

A mobil készülékek számának és típusainak gyarapodása következtében az internetes hozzáférés jelenlegi keretét adó IPv4 protokoll szolgáltatási lehetőségei hamarosan szűknek bizonyulnak. Három cég — a British Telecom, az Ericsson és a SmarTone — most megállapodást kötött, hogy a világon elsőként megkezdik a 2. generációs IPv6 internet protokoll mobil alkalmazásának próbaüzemét, várhatóan 2000 nyarának végén. A fejlesztési munkákat a BT angliai kutatóközpontjában és a SmarTone hongkongi telepén végezték. Az IPv6 számos területen jelent előrelépést elődjéhez képest:

— Jóval szélesebb IP címtartományt kínál a hálózat növekedésével belépő új internetes eszközök számára. (Az IPv4 a jelenlegi fejlődési ütem mellett egy évtizeden belül „betelne”.)

— Differenciáltabb és jobb minőségű internetes szolgáltatást tesz lehetővé, mert képes megkülönböztetni a sürgős adatforgalmat (például az élő hang- és videokapcsolatot) attól, ami „jobban ráér” (e-mail).

— Beépített funkciói vannak az azonosításra, az adatok sértetlenségének ellenőrzésére, a bizalmas információk védelmére. Mindez megkönnyíti az elektronikus kereskedelmi tranzakciók széles körű elterjedését.

D-Link: kombinált csatoló, menedzselhető kapcsoló

Az iparág első kombinált USB hub és 10/100 Mbit Ethernet adapter hálózati eszközét a D-Link cég hozta forgalomba. A DSB-H3ETX az Ethernet csatoló mellett integrált háromportos Universal Serial Bus (USB) hub egységet is tartalmaz, úgynevezett plug-and-play megoldásként könnyen csatlakoztatható bármely szabványos Ethernet, vagy Fast Ethernet hálózathoz. A D-Link szerint az új egység ideálisan alkalmazható hordozható konfigurációkhoz, illetve ún. „dobozon kívüli” installációknál. Várhatóan a kezdő, vagy nem profi PC használók széles tábora értékeli legtöbbre a kombinált adapter előnyeit — nyilatkozta Steven Joe, a D-Link ügyvezető alelnöke.

A D-Link ugyancsak megjelentette DES-3225G jelű, gigabites kapcsolattal rendelkező, 24 portos alapkonfigurációjú 10/100 Ethernet kapcsolóját. A 24-ből két port opciósan kicserélhető 100Base-FX portokkal. A DES-3225G hátoldalán egy további nagy teljesítményű bővítőport található, amely akár egy egyportos 1000Base-SX opciós modult is képes támogatni. Az egység hátlapján található a DES-321G gigabites modulnak szánt port, amely teljes Gbit sebességű csatlakozást tesz lehetővé, akár egy szerverhez, akár a felhasználó hálózatának gerincéhez. A DES-3225G kapcsoló, amely maximálisan 12 MAC címet támogat, a végfelhasználók közvetlen összekötésére és osztott médiaeszközök közötti kapcsoló funkció ellátására egyaránt alkalmazható.

3Com átalakulás: az NIC, LAN/Enterprise üzlet marad

A 3Com március végén megkezdett stratégiai átalakulása május végére hoz változásokat a cég magyarországi tevékenységében. Wojciech Glownia régiómenedzser az április 20-án Budapesten tartott sajtótájékoztatón bejelentette: június elejére eldől a magyarországi iroda új felállás szerinti hovatartozása, a 3Com Magyarország szervezeti struktúrája, és új vezetőjének a személye. „Nem cél, hogy bezárjunk egyetlen helyi irodát is az új régiós szervezeti struktúra miatt” — jelentette ki Glownia, aki megerősítette, hogy a 3Com nem adja fel, sőt fejleszti hálózati interfész kártya (NIC) és nagyvállalati LAN üzletágait. A megújuló 3Com a jövőben hét területre összpontosít. Ezek: nagy sebességű LAN megoldások (maximum 2500 csomópontig), web alapú e-network megoldások, információhozzáférési eszközök és megoldások, vállalati LAN telefónia, szélessávú megoldások, vezeték nélküli technológiák, „vendég alapú” hálózati megoldások.

A tájékoztatón elhangzott, hogy a Dataquest szerint 1999 negyedik negyedévében a 3Com magyarországi piaci részesedése a 10M Ethernet kapcsolók eladásában 51, a 100M Fast Ethernet switch eszközöknél 40, míg a Gbit Ethernet kapcsolók értékesítésében 37 százalékos volt. Az új hálózati eszközök közül áprilisban a SuperStack II Switch 9100 kapcsoló, májusban az 1000Base-T szerver NIC, júliusban pedig a Switch 4000 gerinchálózati kapcsoló kereskedelmi forgalmazását kezdik meg.

Cisco: ADSL router, IP kapcsoló

Április végén dobta piacra 800-as sorozatú hálózati útválasztó eszközcsaládjának legújabb tagját a Cisco. A Cisco IOS technológiájával ellátott Cisco 827 ADSL Router a hálózati ágazat első valódi üzleti osztályú DSL megoldása. Az új router két változatban került piacra. A Cisco 827-es modellt egyetlen 10Base-T interfésszel és egy ADSL csatolóval szerelték fel, míg a Cisco 827-V modell négy analóg telefon interfészt (FXS), egyedi 10Base-T csatolót és egy olyan ADSL interfészt tartalmaz, amely az Alcatel DynaMiTe ADSL chipegyüttest támogatja. GlobalDSL nevű stratégiája révén a Cisco Systems a DSL (Digital Subscriber Line) technológiát a bárhol, bármilyen szolgáltatást igénybe vevő felhasználók számára elérhetővé kívánja tenni. Ennek keretében jelentette be két új IP DSL kapcsolóját (6160 és 6015 típusjellel), amelyek mindegyike multiszolgáltatási alkalmazások széles skálájával rendelkezik, beleértve a legbonyolultabb IP és ATM menedzselt szolgáltatásokat és a multi DSL technológiákat is.

Cabletron: MultiSwitch-szaporulat

Új tagokkal bővült a Cabletron Systems MultiSwitch 900 Ethernet kapcsolócsaládja. A 924TXG és 916TXG egyaránt maximalizálják a MultiSwitch platform előnyeit, hiszen nagyobb sűrűségű 10/100 Ethernet portokkal és moduláris Gigabit Ethernet kapcsolatokkal rendelkeznek, annak érdekében, hogy elérhető áron Layer 2 kapcsolási funkciókat lássanak el, nagy- és kisvállalati hálózati környezetben egyaránt. A MultiSwitch 924TXG 24 darab kapcsolt Fast Ethernet (FE) porttal és egy Gigabit Ethernet (GE) porttal rendelkezik, a 916TXG modellt pedig 16 darab 10/100 kapcsolt FE és két GE porttal képezték ki. Mindkét egységet jellemzi a nyílt technológiájú 5,6 Gbit/secos hátlap, valamint a szabvány alapú, ún. multitechnológiájú kapcsolási képesség. Ez utóbbi révén a Cabletron MultiSwitch platformja könnyen integrálható a különböző szállítóktól származó osztott vagy kapcsolt LAN-okkal.

Chris Birthwell, a MultiSwitch termékek marketingmenedzsere szerint a 900-as család komplett összekapcsolhatóságot biztosít, beleértve a FE, GE, vezeték nélküli, FDDI, ATM, terminálhozzáférési és WAN útválasztási megoldásokat. „Nincs még egy olyan szállító, amely a beruházásvédelemnek ezt a szintjét lenne képes nyújtani” — mondta Birthwell. Mindkét új Cabletron kapcsoló könnyen, plug-and-play módon helyezhető üzembe, 12 ezer MAC címet támogat. Menedzselt berendezésekről van szó, amelyeknél a Cabletron ezen berendezésekért felelős DNPG (Digital Network Products Group) üzletágának ún. „policy” alapú, központi felügyeleti platformról grafikus csatoló révén üzemelő menedzsment alkalmazását használják.

Kovács Attila

Többplatformos üzenetkezelés

Novell Internet Messaging System

A Novell évente megrendezésre kerülő BrainShare konferenciái a lezárult időszak eredményeinek bejelentése mellett a következő fejlesztések irányáról is fellebbentik a fátylat. Az új többnyire persze a korábbi eredményekre épül, így a BrainShare 2000-en (sorrendben a 16. konferencián) sokszor volt szó a folyamatosan fejlesztett címtárra, az NDS-re alapozott termékfejlesztésről.

A DENIM (Directory-enabled Net Infrastructure Model) kialakításával kapcsolatos új termékekről előző havi számunkban már olvashattak. Az új modell lényege olyan termékskála kialakítása (részben a már meglévők továbbfejlesztésével), amely beleillik az internet integráló keretrendszerébe, és alkalmazkodik az e-kereskedelem igényeihez. Az új programfejlesztési modellt három kritikus területnek, a hálózat felügyeletének, a hálózat tartalomszolgáltatási képességének és az internetes portálszolgáltatásnak megfelelően alakították ki. Az új Net Services minden tagja (az iChain, a ZenWorks termékcsalád és az eGuide) szintén illeszkedik ehhez a modellhez.

A bővülő hálózatoknak a rajtuk továbbított üzeneteket nagy tömegben és megbízhatóan kell kezelniük. Ennek megoldására készült a Novell Internet Messaging System (NIMS). Ezt az üzenettovábbítási rendszert az NDS internetes igényekhez igazított verziójának, az eDirectory-nak a képességeire alapozva fejlesztették ki. Mivel ez a címtár képes több platform felhasználóinak adatait egyetlen fastruktúrában kezelni, az ezen alapuló NIMS szintén alkalmas a többplatformos üzenettovábbítási szolgáltatásokra.

Korábban elsősorban a NetWare szerverekre csatlakozott felhasználói programok tudták a NIMS képességeit kihasználni, de már a v2.1 decemberi bejelentésekor elhangzott, hogy ezt hamarosan követni fogja a támogatott platformok körének bővítése a Windows NT és a Unixok irányába. Az idei fejlesztői konferencián már bemutatták a Solarisra és Linuxra elkészült változatokat. A közelmúltban készült el az eDirectory verziója e platformokra is.

Tekintettel a Linux lehetőségeire, ezzel a platformbővüléssel komplex üzenetkezelési rendszert építhetünk ki a legkülönbözőbb fizikai megoldású hálózati szegmensek között, megőrizve az eddig befektetett szellemi és anyagi tőkét. A Linux támogatása az internet lehetőségeinek kihasználásában is kedvező, mert az internetes szolgáltatók körében a Linux egyre népszerűbb.

Az internetes üzenetkezeléstől nem választható el az elektronikus kereskedelmi szolgáltatások kiszolgálása, a hálózathasználatot érintő adatok egységes kezelése. A NIMS ennek megfelelően elsősorban a nagy hálózatok üzemeltetőit célozza meg. Gyors és megbízható levéltovábbítási rendszert kínál az internetszolgáltatóknak és az alkalmazászszolgáltatóknak (ISP és ASP). A kizárólagos központi felügyelet mellett biztosítja az üzenetkezelés, a levélkezelés egyéni kialakítását, tehermentesíti a felügyeletet, és a felhasználóknál is javítja a testre szabhatóságot.

Felhasználói feladat például a jelszavak megváltoztatása, az automatikus levéltovábbítás és a válaszadás feltételeinek kialakítása, a színbeállítás, a preferált nyelv és több más paraméterezés. Kialakítható a védelem a kéretlen reklámlevelek (spam) ellen is. Szintén a felhasználók kényelmét szolgálja a WebMail (a bármely böngészős felületről elérhető postaláda), valamint a MailProxy, amelynek segítségével letölthetők a levelek a szabványos levelezőrendszerekből (Novell GroupWise, Eudora World Mail Server, Microsoft Exchange Internet Mail Service).

A szolgáltatások megvalósításakor a NIMS kezeli a különféle jogosultságellenőrző és biztonsági protokollokat, például a Post Office Protocol 3-at (POP3), az Internet Mail Access Protocol 4-et (IMAP4), a Lightweight Directory Access Protocolt (LDAP) és a Secure Sockets Layert (SSL). Felügyelete — miként az összes rendszermenedzselési tevékenység — elvégezhető a böngészőből. A programról bővebb információ található a Novell honlapján (http://www.novell.com). A Novell által szponzorált MyRealBoxnál (http://www.myrealbox.com) pedig a gyakorlatban is kipróbálható.

Leginkább a vállalati hálózatok belső üzenetkezelését érintik a GroupWise fejlesztéséről szóló információk. A tesztelés alatt lévő új verzió (amelynek kódneve BulletProof, azaz golyóálló) várhatóan ez év végén jelenik meg. Ennek fejlesztése során külön figyelmet fordítanak a más termékeknél is egyre erősebben előtérbe került többplatformos működtetésre.

A végleges verzió az eddigi platformokon kívül Solarison és Linuxon is natívan fut majd. Szintén nyilvános béta stádiumban van a GroupWise mobil változatának fejlesztése, mely az AT&T PocketNet szolgáltatásának felhasználásával készül, és a GroupWise üzleti felhasználói számára mobil kapcsolaton keresztül teszi elérhetővé az adatokat — bármikor és bárhonnan.

Simay Endre István



Biztonság becsomagolva

Norton Internet Security 2000

A Norton névvel fémjelzett termékek nagy karriert futottak be a DOS világában. A Norton Commander talán minden idők legelterjedtebb fájlkezelője, a Norton Utilities pedig szinte a professzionális karbantartás szinonimája. Ez utóbbiból nőtt ki — immár a Nortont felvásárló Symantec fejlesztésében — a Norton SystemWorks programcsomag a 32 bites Windows platformokon végzett munka megkönnyítésére. Hozzá kapcsolódó új elem a beépített technológiáknak köszönhetően gyorsan népszerűvé vált Norton Internet Security 2000 komplex internetes biztonsági csomag.

A Norton Internet Security 2000 (NIS) csomagnak szerves része a Norton Personal Firewall, a Norton Personal Privacy és a Norton Parental Control. A személyes tűzfalat és internetes biztonságot nyújtó alkalmazás 32 bites Windows platformra készült, de leginkább Windows 9x rendszerre ajánlható, mert a csomagban található Norton Antivirus részének tekinthető „Rescue Disk" csomag — ahogy az a programokhoz adott leírásban is olvasható — Windows NT és Windows 2000 esetén nem alkalmas mentőlemez készítésére.

A kereskedelmi (dobozos) termék kipróbálására a Symantec az interneten keresztül bárki által letölthetővé tette a csomag 30 napos próbaverzióját. A 70 MB-os, szabadon hozzáférhető csomagot szerettük volna CD-mellékletünkre is felrakni, hogy olvasóinknak megspóroljuk a nem csekély telefonszámlát meg a macerát, de az offline terjesztéstől a Symantec minden érvelésünk ellenére elzárkózott, így ki-ki magának kell, hogy letöltse azt. Magatartásuk természetesen nem minősíti magát a terméket: más dolog a technológia, és más dolog a „marketing".

Telepítés

A NIS próbaverziójának installálása csak akkor fog elindulni, ha gépünkön telepítve van a hálózati működés támogatása. Felajánlja továbbá a NAV 2000 előzetes telepítését, melynek során mentőlemezek készítését is javasolja, ehhez nálunk 5 darab 1,44 MB-os floppyt kért, de a Norton Rescue Disk rendszer támogatja a ZIP-drive használatát is. (A termék dobozos verziója tartalmaz egy NAV mentőlemezt, amely Win9x-en és NT-n egyaránt használható.)

A NAV a telepítés végén felajánlja a gép újraindítását, amit célszerű is elfogadni, hogy a később telepítendő NIS csak azokat a fájlokat írja felül, amelyeket feltétlenül szükséges. Egyébként telepítéskor a NAV szintén felkínálja az online regisztrálást, amit elfogadhatunk, ha a felajánlott szolgáltatásokat igénybe kívánjuk venni.

A NIS telepítésének forgatókönyve hasonló a NAV telepítéséhez, de szükséges a Microsoft Installer is, amely automatikusan felrakja a programot. (A telepítésmenedzser telepíthető az \MSI könyvtárból.) A próbaverzió is telepíti a Symantec univerzális online frissítőprogramját, a LiveUpdate rendszert, ami különösen a vírusadatbázis frissítéséhez lehet hasznos.

Amennyiben korábban feltették gépükre az (általunk is közreadott) AtGuard programot, akkor arról most le kell mondaniuk, mert jelenlétében a NIS 2000 nem telepíthető ugyanarra a gépre. A Symantec ugyanis megvette az AtGuard licencét, és technológiáját beépítette a NIS 2000-be. Ugyanazt a tűzfalrendszert használhatjuk tehát továbbra is, de egy komplexebb termék részeként (és persze most már sokkal több pénzért).

A gépet a NIS is újraindítja, és mivel alapértelmezésben ez a program együtt töltődik be a Windows-zal, újraindításakor kapunk tőle egy ablakot, amely a próbaverzió esetén mutatja az annak érvényességéből még hátralévő időt. (A NAV azonos ablakával együtt tehát kétszeresen is biztosíthatjuk a program gyártóit arról, hogy értjük a figyelmeztetést.) Ha az automatikus indítás helyett magunk szeretnénk gondoskodni alkalmazásaink indításáról, ezt mindkét programban beállíthatjuk.

Ha a próbaverziót sikerül telepíteni, de ez későbbiekben más Symantec termékek installálását akadályozza, akkor a Regedit programot elindítva meg kell keresni a HKEY_CURRENT_USER\SOFTWARE\, a HKEY_LOCAL_MACHINE\SOFTWARE, illetve a HKEY_USERS\.Default \SOFTWARE bejegyzések alatti Symantec ágat, és a programokra vonatkozó adatokat manuálisan kell törölni, mert a próbaverzió uninstall programja tapasztalataink szerint ezt nem végzi el.

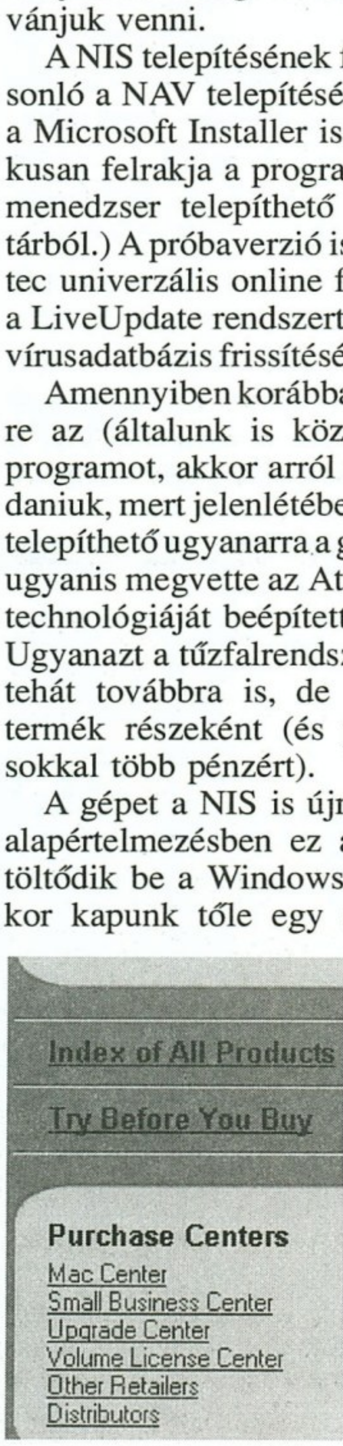
Profilok

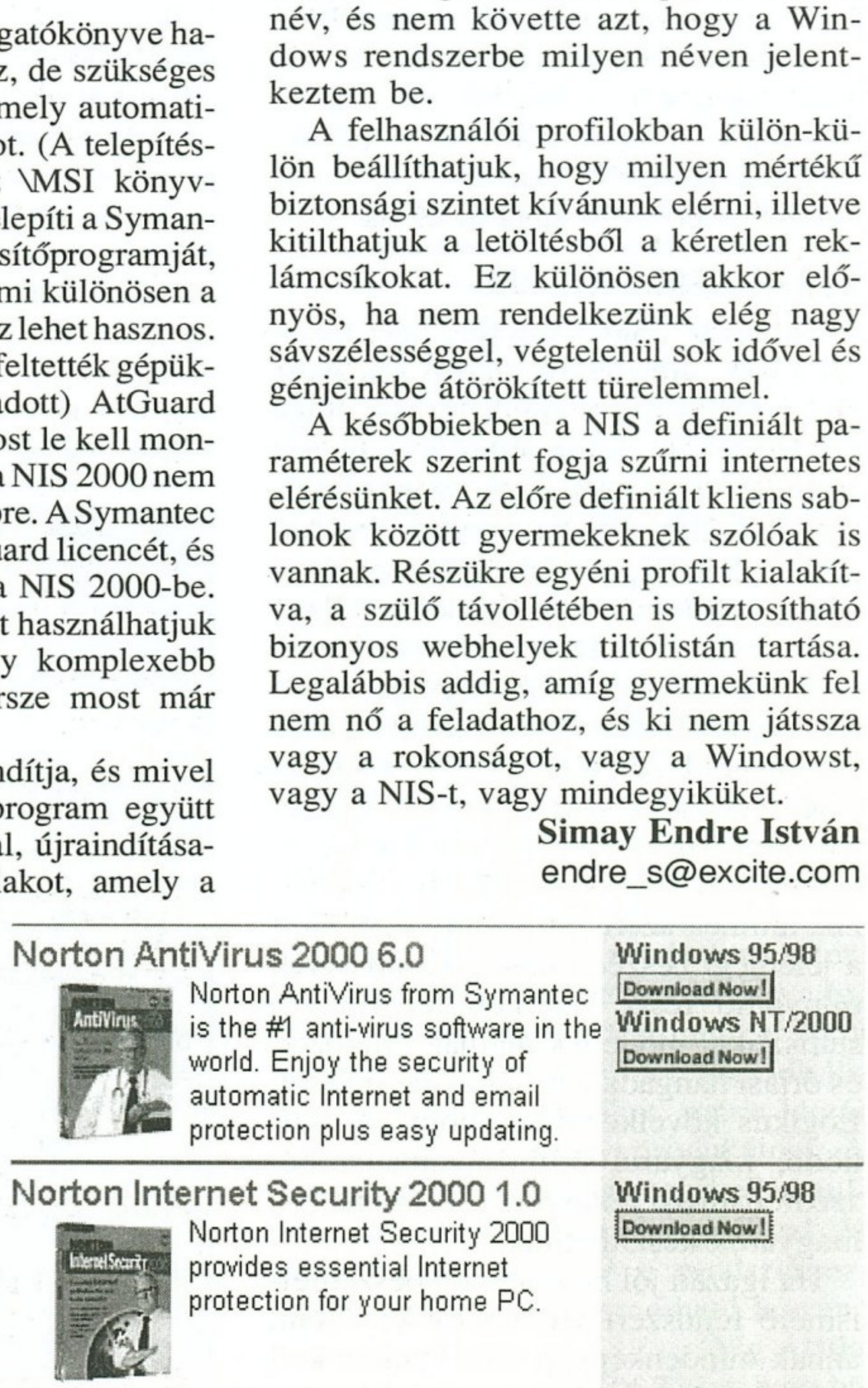
A telepített NIS programegyüttes komplex antivírus + tűzfal rendszerként szolgál. A tűzfal paraméterezését telepítéskor rendszergazdai jogosítvánnyal adhatjuk meg, és új ügyfélazonosítókat is definiálhatunk. Sajátos módon nálam (Windows 95-ön) a próbaverzió tesztelésekor megmaradt a telepítésnél beírt név, és nem követte azt, hogy a Windows rendszerbe milyen néven jelentkeztem be.

A felhasználói profilokban külön-külön beállíthatjuk, hogy milyen mértékű biztonsági szintet kívánunk elérni, illetve kitilthatjuk a letöltésből a kéretlen reklámcsíkokat. Ez különösen akkor előnyös, ha nem rendelkezünk elég nagy sávszélességgel, végtelenül sok idővel és génjeinkbe átörökített türelemmel.

A későbbiekben a NIS a definiált paraméterek szerint fogja szűrni internetes elérésünket. Az előre definiált kliens sablonok között gyermekeknek szólóak is vannak. Részükre egyéni profilt kialakítva, a szülő távollétében is biztosítható bizonyos webhelyek tiltólistán tartása. Legalábbis addig, amíg gyermekünk fel nem nő a feladathoz, és ki nem játssza vagy a rokonságot, vagy a Windowst, vagy a NIS-t, vagy mindegyiküket.

Simay Endre István
endre_s@excite.com





Magyarul diktálni a gépnek

A felismerés nehézségei és eredményei

Az elmúlt 10 év alatt a számítógépek teljesítménye közel ezerszeresére nőtt, a hardverek és szoftverek több generációváltáson mentek keresztül. Ez a fejlődés tette lehetővé, hogy az emberi beszéd gépi felismerése valósággá váljon. Az alkalmazott algoritmusok bonyolultságával együtt javult a felismerés pontossága is. Eljutottunk oda, hogy egy ilyen rendszer használatához bőségesen elegendő egy mai átlagos számítógép kapacitása. „Erős" gép csak a rendszer elkészítéséhez, betanításához szükséges.



Ma a világban számos olyan beszédfelismerő rendszer létezik, amelyet különálló szavak, parancsok felismerésére lehet használni (izolált szavas felismerők). Néhány rendszer arra is alkalmas, hogy komplett dokumentumokat diktálhassunk, a hétköznapi életben megszokott beszédtempóban és hangsúllyal (folyamatos beszédfelismerők). Mindössze egyetlen probléma van velük: többségük csak angolul ért. Néhány még tud németül, franciául vagy más világnyelven, de magyarul egyik sem.

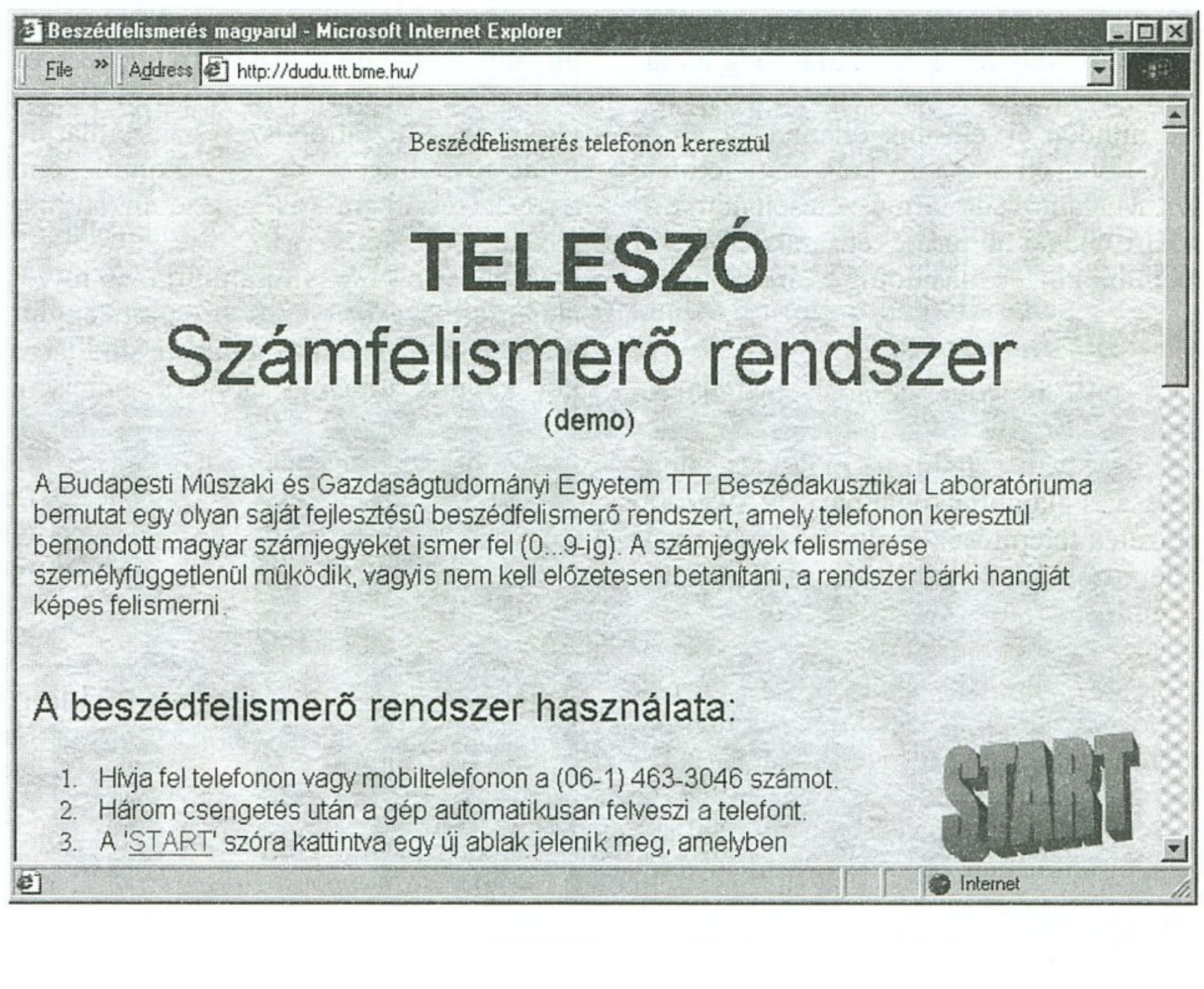
Nyelvi specialitások

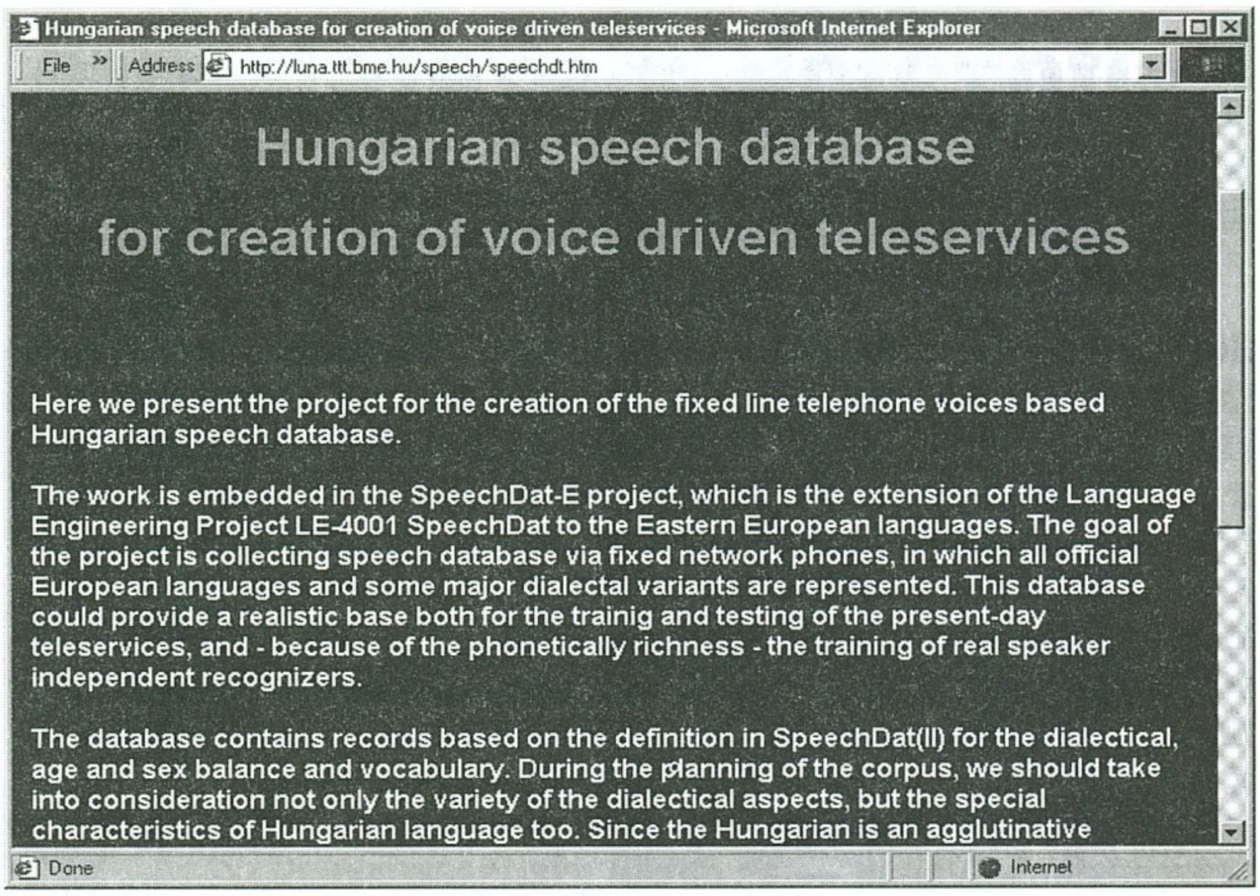
Túl azon, hogy a magyar nem tartozik a nagy világnyelvek közé, ennek az elhanyagoltságnak egyik fő oka, hogy nyelvünk sok szempontból lényegesen különbözik a nagy indoeurópai nyelvektől. A magyarban nagyságrendekkel több a lehetséges szóalakok száma a nyelv agglutináló (toldalékos) jellege miatt. Ez eleve lehetetlenné teszi, hogy a felismerés legkisebb egysége a szó legyen, ami számos külföldi rendszerben megoldható, a magyarban a szónál kisebb egységekben kell gondolkodni. Emellett a hangzók és a hangkapcsolatok statisztikája is jelentősen különbözik, aminek azért van jelentősége, mert a jelenlegi beszédfelismerő rendszerek túlnyomó része (ha nem az összes) statisztikai modellek alapján dolgozik, és óriási hangadatbázisokat használ fel. Logikus következmény, hogy jól működő, magyarul értő folyamatos beszédfelismerő rendszert elsősorban mi, magyarok készíthetünk.

Ha igazán jól használható beszédfelismerő rendszert szeretnénk készíteni, annak mindenképpen több szinten kell a felismerést elvégeznie. A gyakorlatban ez egy akusztikai és egy nyelvi szintet jelent, persze mindkettő állhat több alszintből is. Az akusztikai szint közvetlenül az időjelet, vagy (ami általánosabban elterjedt) annak valamilyen frekvenciatartománybeli változatát elemzi (spektrum, cepstrum, sávszűrők). Ezen a szinten elég sok hibát okozhat a kiejtési pontatlanság és a háttérzaj. A hibák jó része azonban kiküszöbölhető, ha a felismerés során a nyelvi szabályokat is figyelembe vesszük. Ha például egy szó egyaránt állhat a „h á r o m" és a „h á r a m" fonémák sorozatából, továbbá tudjuk, hogy ennek egy számnak kell lennie, mindjárt egyértelmű lesz a válasz. Magasabb nyelvi szinten, ha egy mondat első szava egyforma valószínűséggel lehet „Lobog" vagy „Robog", akkor sokat segíthet a döntésben, ha tudjuk, hogy a mondat többi részében előfordul „a tűz".

Hazai fejlesztés

A Budapesti Műszaki és Gazdaságtudományi Egyetem (BME) távközlési és telematikai tanszékének beszédakusztikai laboratóriumában immár több mint 10 éve foglalkozunk beszéd-




Hungarian speech database for creation of voice driven teleservices - Microsoft Internet Explorer

File » Address http://luna.ttt.bme.hu/speech/speechdt.htm

Hungarian speech database

for creation of voice driven teleservices

Here we present the project for the creation of the fixed line telephone voices based Hungarian speech database.

The work is embedded in the SpeechDat-E project, which is the extension of the Language Engineering Project LE-4001 SpeechDat to the Eastern European languages. The goal of the project is collecting speech database via fixed network phones, in which all official European languages and some major dialectal variants are represented. This database could provide a realistic base both for the trainig and testing of the present-day teleservices, and - because of the phonetically richness - the training of real speaker independent recognizers.

The database contains records based on the definition in SpeechDat(II) for the dialectical, age and sex balance and vocabulary. During the planning of the corpus, we should take into consideration not only the variety of the dialectical aspects, but the special characteristics of Hungarian language too. Since the Hungarian is an agglutinative

Done Internet

felismeréssel. Többek között a nagy beszédadatbázisok létrehozásában, azok hanganyagának statisztikai elemzésében, a betanításhoz történő előkészítésében, valamint különböző felismerési módszerek kidolgozásában értünk el jó eredményeket. Jelenleg egy személy- és szövegfüggetlen, a folyamatos beszédet felismerő rendszeren dolgozunk, amely mostani állapotában az irodai körülmények között (>15-20 dB jel/zaj mellett) bárki által mikrofonon keresztül bemondott tetszőleges magyar mondatokat 70-80%-ban képes felismerni.

A rendszert leginkább szövegszerkesztő programok kiegészítéseként, a billentyűzet helyettesítésére lehet majd használni. A jelenlegi pontosság még nem teszi lehetővé, hogy diktálhassunk neki, de más feladatok ellátására kisebb-nagyobb megszorításokkal alkalmassá tehető. Ha a szótárkészletet leszűkítjük, és nem engedjük meg tetszőleges szavak használatát, akkor a felismerési arány 90%-nál is nagyobb lehet. Az így létrejövő felismerő program ún. dialógusrendszerek kialakítására is alkalmas.

Párbeszéd

Dialógus rendszereknek nevezzük az ember és a gép közötti kétirányú kommunikációt megvalósító rendszereket. A beszéd útján történő dialógusban a felhasználó egy-egy szóval, rövid mondatokkal válaszol a gép által feltett kérdésekre, illetve ő is kérdezhet a géptől. Ugyanaz a kérdés többféleképpen is feltehető, nem kell mindig ugyanazt a formát használni. (Például a 'Mennyi az idő?' és a 'Hány óra van?' kérdés variálható.)

Bár a szótárkészlet ilyen esetekben elég korlátozott, viszonylag változatosan fogalmazhatunk a gépnek. Ilyen alkalmazás lehet például a menetjegyrendelés, amikor megmondjuk a gépnek, hogy melyik nap, melyik városba, milyen időpontban szeretnénk utazni.

A rendszer visszakérdezhet, hogy ablak mellé, dohányzóba, illetve első osztályra szeretnénk-e a jegyet. Jó néhány hasonló példát említhetnénk még, a mozijegyrendeléstől kezdve a banki információs rendszereken keresztül egészen az autóba beépített műholdas helymeghatározó berendezéssel kombinált útvonallekérdező rendszerekig, ahol az aktuális forgalmi helyzetet figyelembe véve kaphatunk információt a kívánt úticél leggyorsabb eléréséhez.

Teleszó

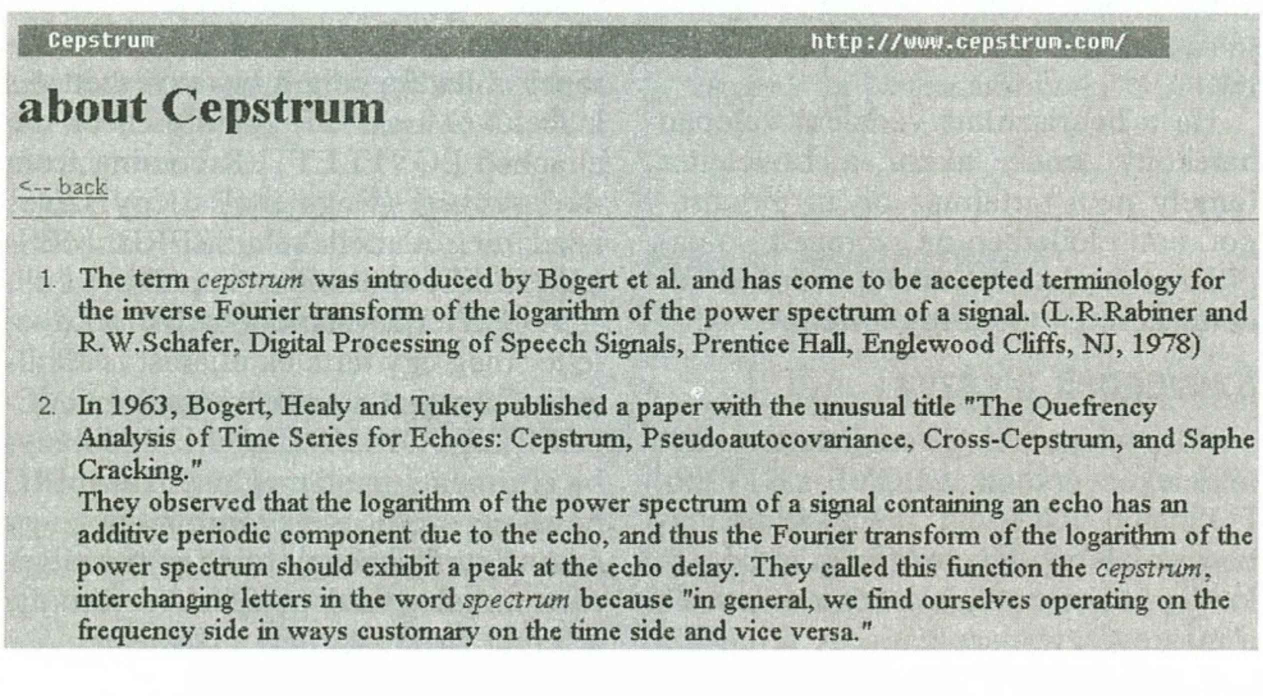
Az előbbivel párhuzamosan fejlesztjük Teleszó nevű rendszerünket, amely telefonon keresztül bemondott szavakat (egyelőre még csak számokat) képes szintén beszélőfüggetlenül felismerni. A telefonos felismerés jóval nehezebb a mikrofonosnál, elsősorban az átviteli sáv korlátozott volta miatt, de a nagyobb zaj is gondot okoz. Mindezek ellenére a felismerés biztonsága 95% körüli, a tévesztések nagy része pedig elutasító jellegű, vagyis a fel nem ismert szót nem másik számjeggyel téveszti össze, hanem mindegyiktől teljesen különbözőnek találja.

A Teleszó rendszert bárki kipróbálhatja, aki ellátogat a http://dudu.ttt .bme.hu weboldalra. Az ott megadott telefonszámot tárcsázva a program a bemondás alapján felismert számjegyeket az interneten keresztül kiírja a képernyőre. A felismerés gyorsasága gyakorlatilag csak a hálózat sebességétől függ, legfeljebb emiatt kell várni néhány másodpercet. Ez elsősorban a modemes kíváncsiskodóknál fordulhat elő (ráadásul két telefonvonalra van szükség a program kipróbálásához).

A rendszer egyelőre kísérleti jelleggel üzemel, folyamatos fejlesztés alatt áll. Célunk egy olyan szófelismerő modul elkészítése, amely több operációs rendszer alatt is működik és tetszőleges programba beágyazható. Egyszerre néhányszor tíz szó felismerésére lesz alkalmas, de a szótárkészletet tetszőlegesen lehet majd bővíteni, egyszerűen a felismerendő szavak leírásával.

A beszédfelismerő rendszerek elterjedéséhez a legfőbb követelmény a további intenzív fejlesztési munka. Ha a meglévő programokból a fejlesztők kiszedik a programhibákat, máris látványosan jobb eredményt produkálnának. A felhasználóknak pedig meg kell tanulniuk olyan stílusban, hangsúllyal és hanglejtéssel beszélni a rendszerhez, hogy az könnyen felismerhető legyen.

Vig Attila
vig@ttt-202.ttt.bme.hu


Cepstrum http://www.cepstrum.com/

about Cepstrum

<-- back

1. The term *cepstrum* was introduced by Bogert et al. and has come to be accepted terminology for the inverse Fourier transform of the logarithm of the power spectrum of a signal. (L.R.Rabiner and R.W.Schafer, Digital Processing of Speech Signals, Prentice Hall, Englewood Cliffs, NJ, 1978)

2. In 1963, Bogert, Healy and Tukey published a paper with the unusual title "The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudoautocovariance, Cross-Cepstrum, and Saphe Cracking."
They observed that the logarithm of the power spectrum of a signal containing an echo has an additive periodic component due to the echo, and thus the Fourier transform of the logarithm of the power spectrum should exhibit a peak at the echo delay. They called this function the *cepstrum*, interchanging letters in the word *spectrum* because "in general, we find ourselves operating on the frequency side in ways customary on the time side and vice versa."

Változatok a szerelemre

Mi az oka védtelenségünknek?

Rövid időn belül másodszor csapott le az informatikai társadalomra egy igen hatékonyan működő féregprogram, világszerte komoly problémákat okozva. E sorok írásakor a sebek nyalogatása és az áldozatok összeszámlálása még folyamatban van. Az első csapás, a Melissa után azt gondolhattuk volna, hogy az elég erős figyelmeztetés volt, és mindenki jobban odafigyel majd erre a potenciális veszélyre. Különösen mivel az írott és elektronikus sajtó is minden korábbit felülmúló terjedelemben foglalkozott a témával. Újfent rá kellett azonban döbbenni, hogy sokkal jobban ki vagyunk szolgáltatva a modern számítástechnikának — és önnön gyarlóságunknak.

A férgek olyan elképesztő sebességgel képesek az elektronikus levelezőrendszereken keresztül terjedni, hogy megjelenésük teljesen új feladat elé állítja a víruskeresők fejlesztőit.

Korábban a hagyományos programvírusok elég lassan jutottak át egyik gépről a másikra, a bootvírusok pedig csak „gyalogos" sebességgel terjedhettek, lévén hogy fertőzött floppyval kellett közlekedni a számítógépek között. A makróvírusok már valamivel gyorsabb szaporodást tettek lehetővé, mert Word dokumentumokat levélmellékletként küldözgettek a cégek belső hálózatán vagy kifelé, de még így is napokig eltartott egy szélesebb körű fertőzés kialakulása.

Versenyfutás az idővel

A vírusirtók ehhez alkalmazkodva viszonylag kényelmes munkatempóban dolgozhattak, havonta (a szorgalmasabbja hetente, az F-Prot sztahanovistái naponta) adtak ki frissítést a vírusadatbázishoz. Persze ha valahol komolyabb fertőzés tört ki, akkor a beküldött minta alapján néhány órán belül, de legkésőbb másnap jött a frissítés. Nagy baj ennyi idő alatt nem történhetett, egy-két óra vagy egy nap alatt legfeljebb néhány újabb gép fertőződött meg, azokat pedig utána körbejárhatták és letisztíthatták a rendszergazdák.

A korszerű féregprogramok hipergyors terjedésükkel felborították ezt a rendet. Már a Melissa is, most pedig a Loveletter bebizonyította, hogy sokkal gyorsabban kell megcsinálni a vírusadatbázis frissítését, ugyanis ezek a férgek percek alatt szétterjedhetnek, és a fertőzés nem korlátozódik a kiindulási pont környezetére, hanem szó szerint az egész világra kiterjednek. Mire tehát a víruskereső frissítése megérkezne (persze e-mailben), hogy gátat szabjon a féreg terjedésének, az már annyira leterheli a levelezőrendszert, hogy célszerűbb átmenetileg leállítani. De ha működik is a levelezés, a féreg a gépek százain, ezrein terjedt el, ami rendkívül megnehezíti az irtását.

Be kell vallani, hogy a vírusirtók jelen pillanatban még nem rendelkeznek megfelelő megoldással erre a problémára. A védelmek többsége a már ismert kártevők detektálására koncentrál, az újak felismerése nem tekinthető megbízhatónak. Majdnem minden vírusvédelmi termék rendelkezik ugyan heurisztikus analizátorral, amely a programkód jellege alapján próbálja meg eldönteni, hogy az ismeretlen jövevény vírusszerű-e, de ezek még korántsem tökéletesek, és sok a vakriasztás is.

Ha a heurisztikus védelem valóban hatékony lenne, akkor a Loveletter (amely nem tartalmaz olyan újdonságot, ami elődeiben ne szerepelt volna) pillanatok alatt fennakadt volna a védelmeken. De nem így történt.

Szövegnek álcázva

A Loveletter féreg e-mail üzenet mellékleteként érkezik, a LOVE-LETTER-FOR-YOU.TXT.vbs nevű állományban, amelyre a levelezőprogramból rákattintva életre kel. A kettős kiterjesztés a megtévesztést szolgálja. A Windows 95/98/NT rendszerek ugyanis alapértelmezésben nem írják ki a fájlnév kiterjesztését, így ezt is úgy mutatják, mintha egy TXT kiterjesztésű (tehát ártalmatlan) szöveg lenne.

Ez az átverés elég jól működik, hiszen egy (magyarra fordítva) NEKED-SZÓLÓ-SZERELMESLEVÉL.TXT fájl nem kelt gyanút, inkább valami viccnek ígérkezik, s az ilyenekbe már csak kíváncsiságból is sokan bele akarnak nézni. Persze a figyelmesebb hozzáértőknek feltűnhet, hogy egy szövegállomány nem szokott VBScript ikonnal megjelenni, de a felhasználók nagy tömege minden további nélkül bedől a cselnek.

Trójai funkció

A Loveletter ezt követően több példányban bemásolja magát a megtámadott számítógépre, majd a Windows rendszer konfigurációs állományaiban olyan változtatásokat hajt végre, hogy minden további rendszerindításkor aktivizálódjon. Ezzel párhuzamosan a SkyiNet fülöp-szigeteki internet szolgáltató egyik honlapjáról megkísérel letölteni egy WIN-BUGSFIX.exe nevű állományt, amely valójában egy jelszólopó program. Sikeres letöltés esetén azt is úgy jegyzi be a gépen, hogy rendszerindításkor elinduljon és összegyűjtse a Windows memória cache-ében lévő jelszavakat, majd e-mailben elküldje azokat a megadott címre a Fülöp-szigeteken.

Mindezek után a Loveletter megkísérli elindítani a Microsoft Outlook levelező programot. Ha az Otlook az adott számítógépen telepítve van, akkor az ActiveX programozási felületét felhasználva a felhasználó összes címjegyzékében szereplő valamennyi e-mail címre kiküldi önmagát egy példányban.

A kiküldött levelek melléklete maga a féregprogram a már említett megtévesztő fájlnévvel. A Subject (tárgy) mezőben az ILOVEYOU „felirat" szerepel, a levél pedig a hevenyészett (és kisbetűvel kezdődő) „kindly check the attached LOVELETTER coming from me" szöveg. (Nagyjából: „légy szíves nézd meg a mellékelt SZERELMES-LEVELET, amit tőlem kaptál".)

A fenti terjedési módon kívül a Loveletter még egy tartalék eljárást is alkalmaz. Önmagát némi körítéssel LOVE-LETTER-FOR-YOU.HTM állományba is kimenti, majd megkísérli az mIRC nevű népszerű csevegőprogramot úgy átkonfigurálni, hogy minden beszélgetőcsatornába való belépéskor elküldje a férget tartalmazó állományt.

Variánsok

A Loveletter csak Microsoft Windows operációs rendszerek alatt életképes, ott is csak akkor, ha futtató környezete, a Windows Scripting Host telepítve van. Ezek tipikusan a Windows 98, a Windows NT és aWindows 2000, illetve a Windows 95 is, ha azon az Internet Explorer 5.0 (vagy egy későbbi változat) telepítve van. A sikeres továbbterjedéshez szüksége van továbbá a Microsoft Outlook vagy az mIRC program jelenlétére.

Alig egy héttel az alapváltozat megjelenése után a DataFellows weblapján már 23 variáns részletes elemzését készítették el, a Symantec szakértői pedig 29 változatot soroltak fel. Mi a variánsok gyors színrelépésének az oka? Leginkább az, hogy rendkívül könnyű a módosítás, a féreg ugyanis forráskódban terjed, a szöveges VBScript állomány bármilyen szövegszerkesztővel átírható, és máris kész a mutáció. Az első változatok éppen ezért többnyire primitív módosítások voltak, legtöbbször csak a férget terjesztő e-mail címsorát, illetve a levél tartalmát változtatták meg. Motiválhatta a variánsok gyártóit az is, hogy minimális kreativitással a figyelem középpontjában érezhették anonim produktumukat. A későbbi változatok egyre veszedelmesebbek lettek.

Károkozás

Már az incidens után pár nappal több milliárd dolláros okozott kárról lehetett hallani. Ez nem közvetlen károkozásból származik, hiszen az eredeti Loveletter felülír ugyan egyes állományokat (VBS, VBE, JS, JSE, CSS, WSH, SCT, HTA kiterjesztésűeket, illetve JPEG, MP2 és MP3 fájlokat), de ezekből aligha jön össze ekkora summa.

A férgek általában nem direkt rombolásukról híresek, hanem terjedésük viharos sebességéről: eldugaszolják, lebénítják a levelezőrendszereket. Mind a Melissa, mind a Loveletter esetében hosszú órákra „padlóra kerültek” világcégek és kormányzati szervek levelezőrendszerei is.

Az ilyen leállások kára közvetlenül talán nem is mindig mérhető, de hatásukra csökkenhet a bizalom ezen cégek iránt, ami akár a részvényárfolyamokra is kihat.

A Loveletter esetében enyhe pánikreakciókat is meg lehetett figyelni. A fertőzéstől félve feleslegesen zártak le egyes levelezőrendszereket, aminek végeredményben ugyanolyan hatása volt, mintha a féreg okozott volna torlódást: leállt a levélforgalom.

Kasszandra-jóslataink

Az Új Alaplap nagyon sok írása „megelőzi korát”. Részben olyan információkkal, melyek az informatikai szakemberek későbbi munkájához szolgálnak támpontul, részben a fejleményeket jó előre „megjósló” elemzésekkel. Törzsolvasóink tudják, hogy nyugodtan elővehetik 5-10 évvel ezelőtti számainkat is, azok tartalma meglepően időtálló. Ezzel mi nem nagyon szokunk kérkedni, de most kivételt teszünk. A „korszellem” ezt követeli.

Az ILOVEYOU vírus (féreg) kapcsán érdemes elővenni Szappanos Gábornak az elmúlt évben megjelent írásait a Melissa — meg a többi, e-mailben terjedő kártevő — „létalapjáról” és annak belátható (vagy beláthatatlan) következményeiről. A probléma sűrítményeként álljon itt most két hosszabb idézet:

„Sajnos a Microsoft nem tanult az előzményekből: a VBScript nyelv és az interpreter is teljes hozzáférési lehetőséget kap a böngészőt futtató számítógép fájlrendszeréhez. Pedig csak egy kicsit kellett volna tanulmányozniuk a hasonló célokat szolgáló JavaScript megoldását. Ott ugyanis a teljes fájlkezelést kihagyták. Ez persze korlátokat szab a webprogramozók szárnyaló fantáziájának, ezért a Microsoft vélhetően úgy döntött, hogy a könnyebb programozhatóság (és ennek révén a webfejlesztők elcsábítása) érdekében nem szab ilyen korlátokat, kiterjeszti a programozási lehetőségeket. Hát nem is kellett sokáig várni arra, hogy ezt a védelmi lyukat kihasználó első vírusok megjelenjenek. Legfeljebb az a meglepő, hogy nem bukkantak fel még hamarabb. Erre egyetlen magyarázatot látok: az igazán komoly vírusírók méltóságon alulinak találták a scriptvírusokkal való foglalkozást („sas nem kapdos legyeket”), a kezdőknek pedig egyszerűen nem jutott eszükbe. Mindezt alátámasztja, hogy amikor az első példány megjelent (és ezzel mindenki számára világossá vált, hogy akár VBScriptben is lehet vírust írni), akkor két hét alatt egy tucat új scriptvírust gyártottak.

... Az Interneten keresztül történő fertőzés a kezdő vírusírók fantáziáját megmozgatja, és az eddig megjelent példányok alapján egy kutyaütő is össze tud hozni 5 perc alatt egy új változatot. Ezt még jobban megkönnyíti, hogy ezek a vírusok szabadon módosítható forráskód formájában terjednek. A VBScript-vírusok akár a makróvírusoknál is nagyobb karriert futhatnak be, pedig ez utóbbiakból a gyenge első fél év után 3 év alatt több mint 3000-et készítettek. Elég sötét jövőkép tárul fel előttünk, és ezt nem a vírusszakértők kötelező pesszimizmusa mondatja, hanem a vírusírás tendenciáiból leszűrhető tapasztalat.” *(Új Alaplap, 1999. február)*

„A Microsoft koncepciója szerint megalkotott integrált irodai alkalmazáscsomag biztonsági lyukai ugyan sokszor terítékre kerültek, de kiküszöbölésükre olyannyira nem történt semmi, hogy lám most az Office 2000 még forgalomba sem került, de a hiányosságait meglovagoló vírus már itt ül a nyakunkon.

Ez a mostani, lavinaszerű fertőzés mindenki számára a gyakorlatban is bebizonyította, hogy a jelenlegi Word–Excel–Outlook rendszer biztonságtechnikai szempontból használhatatlan, a Melissa típusú támadásokkal szemben konstrukciójából adódóan teljesen védtelen. Egy virulens fertőzés órák alatt végigseperhet igen nagy és erős cégeken is, mielőtt még bárkinek alkalma lenne beszerezni és lefuttatni a naprakész víruskeresőt.

Amikor az Internet Worm felbukkant, többek között azért nem terjedt el nagyobb mértékben, mert az Interneten alkalmazott operációs rendszerek heterogénabbak voltak. Gondoljunk azonban bele, mi lesz, ha olyan operációs rendszer uralja a világ számítógépeinek túlnyomó többségét, amelyben „hemzsegnek” a biztonsági rések.” *(Új Alaplap, 1999. május)*

A szerk.

Hogyan levelezzünk?

Az e-mailen keresztül terjedő férgek ellen elvileg nagyon egyszerű a védekezés: nem szabad elindítani a levélhez mellékelt programokat, amíg azok eredetéről nem kapunk hiteles megerősítést. Ha tartanánk magunkat ehhez az egyszerű szabályhoz, esélye sem lenne a Loveletter típusú kártevőknek. Miért dőlnek be mégis újra meg újra olyan sokan? Azért, mert a férget tartalmazó levél olyan feladótól érkezik, akivel levelezésben állunk (hiszen a küldő címjegyzékében szereplő emberekhez jut el a féreg), ezért automatikusan biztonságosnak tekintjük a levél tartalmát. Legyünk tehát körültekintőbbek! Levélmellékleteket a körülmények ismerete nélkül ne futtassunk le, még ha a „feladó” az ismerősünk is.

Szappanos Gábor

Teszt a lelke mindennek

Vírusirtók összehasonlítása

Már megszokhattuk, hogy a hamburgi Virus Test Center dolgos munkatársai nagyjából félévente elvégzik a szakma legszélesebb körű víruskereső tesztjét. Tették ezt idén februárban is, de ezúttal csak a makróvírusokra koncentráltak. Minden eddiginél nagyobb anyagon, 4525 különböző makróvírust tartalmazó gyűjteményükön tesztelték a programokat.

Program	Talált vírus	Találati arány
AntiVir 5.21	3888	85.9
Avast 7.2	4239	93.7
Antivirus Pro	4522	99.9
Command Software AV	4525	100.0
Dr.Web	4452	98.4
F-Prot	4525	100.0
F-Secure Antivirus	4521	99.9
Inoculan 4.53	4512	99.7
Norton Antivirus	4408	97.4
NOD	4500	99.4
Norman Virus Control	4521	99.9
Panda Antivirus	4522	99.9
NAI Scan	4525	100.0
Sophos Sweep	4453	98.4

A hamburgi Virus Test Center legutóbbi nagy felmérésének részletes kiértékelése az ftp://agn-www.informatik.uni-hamburg.de/vtc címen mindenki számára hozzáférhető, e helyütt csak mazsolázgatunk az eredményekből. Három kategóriában versenyeztek: DOS programok, Windows 98 alatti keresők és a Windows NT keresők tudását mérték össze (a legtöbb fejlesztőcsapat mindhárom kategóriában indult).

Részletesen csak a DOS alatti keresők végeredményét ismertetjük a mellékelt táblázatban. Gyakorlatilag minden fejlesztőműhely esetében ugyanaz a víruskereső motor pöfög ugyanis a több operációs rendszeren is, csak a köréjük pakolt felépítmény más. Minél komplikáltabb az operációs rendszer, annál komplikáltabb a felépítmény, és annál több rejtett hiba kerülhet bele. Emiatt az az általános vélekedés, hogy a víruskereső motorok teljesítményét legtisztábban a DOS változatoknál lehet lemérni.

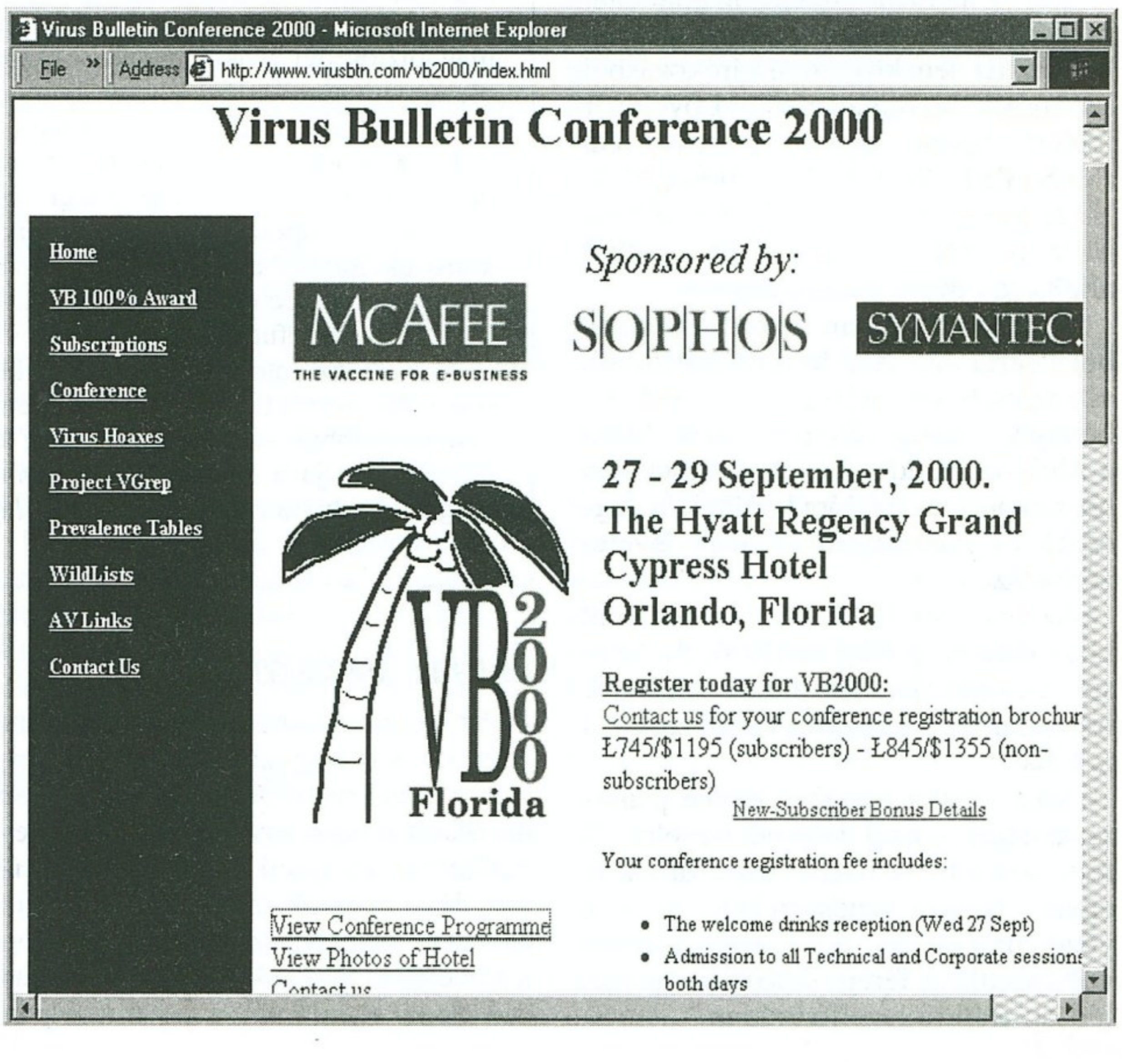
100%-os keresőnek bizonyult a Command Software Antivirus (amelynek szívében az F-Prot víruskereső motorja dobog), az F-Prot és ismét a régi fényében tündöklő Scan.

A Windows NT alatti keresők közül hibátlanul vizsgázott a Command Software Antivirus, az F-Prot for Windows, az F-Secure (amelyben szintén az F-Prot motorja fut) és a Scan. A Windows 98 alatti keresők közül a Command Software Antivirus, az F-Prot for Windows, az F-Secure, az F-Win és a Scan volt a legjobb.

A vizsgált programok közül egyedül a Scan produkált tökéletes eredményt a hamburgiak legutóbbi három tesztje során. Ezt egyrészt az magyarázza, hogy elég jó szakértőik vannak, másrészt pedig jó vásárt csináltak, amikor megszerezték a Dr.Solomon's Antivirus Toolkitet, tokkal vonóval együtt. (Rosszindulatú pletykák szerint a jó eredményben szerepe lehet annak is, hogy a NAI jelentős összeggel támogatja a Virus Test Centert, de ezt terjeszthetik pusztán irigységből is.)

Az F-Prot alapú víruskeresők jó szereplése nem meglepetés, ez a víruskereső már régóta stabilan az élvonalban van. Az F-Win teljesítménye csak azoknak meglepetés, akik nem tudják, hogy ezt a programot már az első makróvírusok megjelenésekor elkezdték fejleszteni, és teljesen erre a víruscsoportra specializálták. Igazán meglepő az, hogy a program Windows NT alatt hibázott, nem ismert fel 3 vírust. Ez ismét arra utal, hogy a vírusfelismerési rátát nem kizárólag a víruskereső motor határozza meg, hanem a köré épített felépítménynek és az operációs rendszernek is van némi szerepe.

Szappanos Gábor
szapi@scl.sztaki.hu

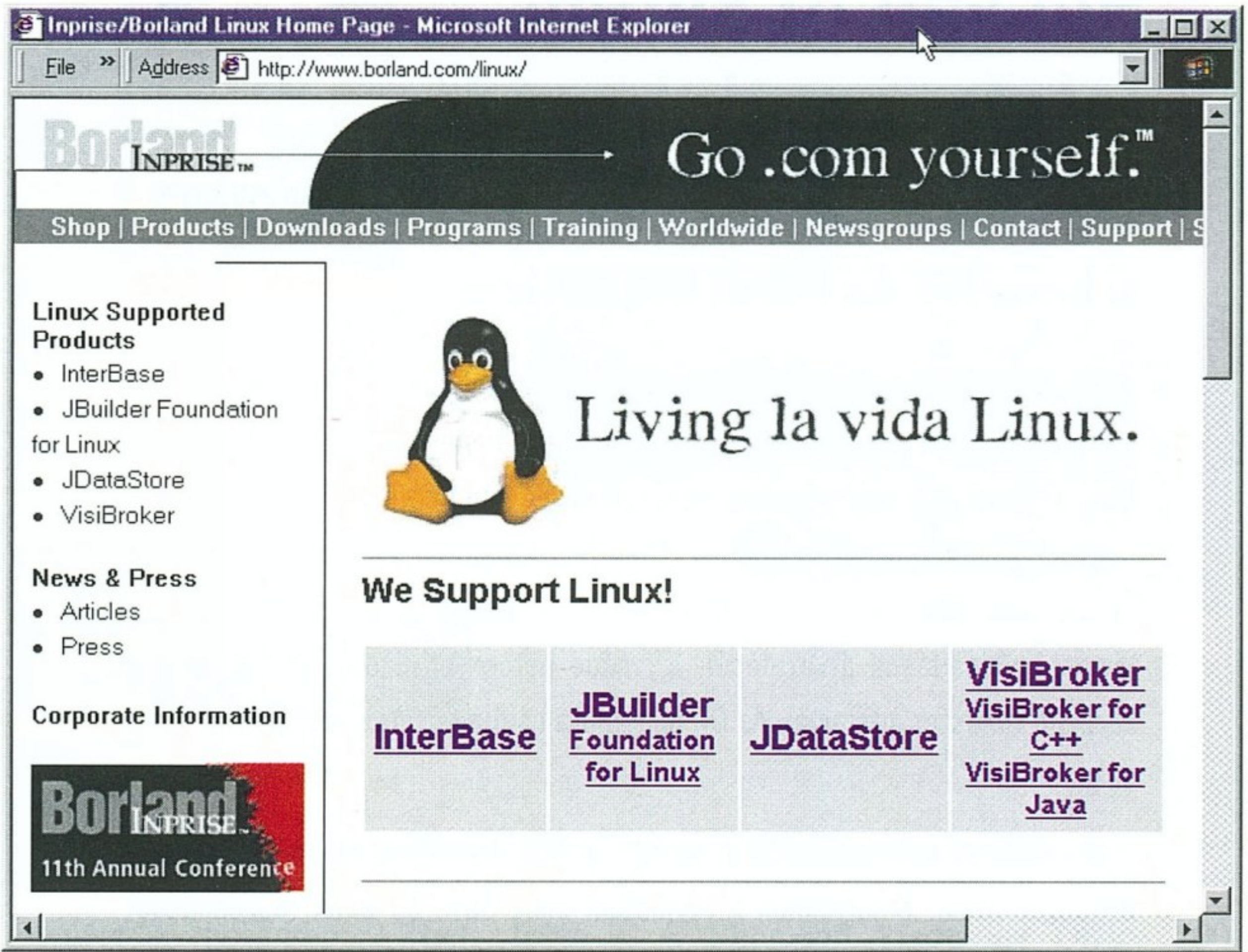


INFO 2000

A legnagyobb tavaszi számítástechnikai rendezvény kétségtelenül az INFO 2000 volt. Mindamellett ez a kiállítás egy negatív trend újabb állomásaként vonulhat be a szakmai krónikába. Bizonyára sokan emlékeznek még rá, hogy néhány évvel ezelőtt 11 jelentős informatikai cég közösen nyilatkozott arról, hogy évente két vásári részvétel (Ifabo és Compfair) a magyar piacot figyelembe véve túl sok, ezért úgy döntöttek, hogy a kettő közül a tavaszi Ifabót részesítik előnyben. Idén nem volt semmiféle kollektív nyilatkozat, csupán a Compaq kivételével szép csendben távol maradtak a legnagyobb, a piacot döntően befolyásoló számítástechnikai cégek. Ha a korábbi érvelés logikáját követjük, akkor nekik a magyar piac most már évente egyetlen informatikai vásáron való részvételt sem tesz indokolttá, kifizetődővé.

Mindebből persze még nem szabad olyan következtetésre jutni, hogy a számítástechnika leszálló ágba került. 2000 közepén (az állítólagos holtszezonban) például informatikai konferenciák tucatjai zajlottak és zajlanak le. Emellett a kommunikáció megannyi csatornája szuggerálja, hogy már itt vagyunk az informatikai társadalom kapujában (elektronikus kereskedelem, internetezés, mobil telefónia stb.). Ha nem is kell minden nyilatkozatot szó szerint értelmezni, az informatika bizonyosan továbbra is az emelkedő ágban van, csak a hozzá kapcsolódó szakmai ismeretek terjesztéséről és az alkalmazandó piaci módszerekről alkotott felfogásban történtek bizonyos változások, amelyek logikáját nem mindig sikerül követnünk. (A választ kereső eszmefuttatások előtt nyitva áll Nyílt tér és Visszacsatolás rovatunk!)

Az INFO arra is alkalmat adott, hogy kibontakozzon az „ott is vagyok, meg nem is" mozgalom. Az átmenetileg nehéz anyagi helyzetbe kerülő cégek ismert módszere, hogy nem bérelnek saját standot, de azért valahogy igyekeznek jelen lenni. Érthető tehát, hogy a Microsoft is rábízta jelenlétét a Windows feliratú reklámzacskókat osztogató mobil leányzókra, illetve a Windows alá fejlesztő vagy ilyen programokat forgalmazó kiállítókra. Ezen legfeljebb dünnyögünk egyet. Arra viszont már felkaphatjuk a fejünket, hogy versenysemlegességet képviselni hivatott Miniszterelnökség Közbeszerzési és Gazdasági Igazgatósága standján több más, a közbeszerzési folyamatokban érdekelt, de kiállítóként távol maradó cég szintén képviseltette magát, szóróanyaggal, felirattal...



A nagy távközlési vállalatok jelenléte nélkül az idei INFO elég soványra sikeredett volna. Szerencsére az új húzó ágazat valamelyest betölti azt az űrt, amely a „klasszikus" számítástechnika kivonulása nyomán keletkezne. Persze nem mindig kiszámítható módon. A vásári mászkálásban elfáradt látogatók például nagy örömmel fedezhettek fel egy növényekkel, padokkal berendezett tágas szabad területet az „A" pavilon azon részén, ahol a katalógus a Matáv standját jelölte. Nyugodtan ki lehetett volna tenni egy táblát az alábbi felirattal: „Ez a kellemes pihenőhely a 'last minute' visszalépő Matáv jóvoltából áll az Önök rendelkezésére."

Távolmaradó kiállítók ide vagy oda, idén is több 10 ezer ember vette a fáradságot (és a belépőjegyet), hogy az informatikával kialakult kapcsolatát friss tartalommal töltse meg. Lehet persze, hogy valami új koncepció jegyében a vásárlátogató — legyen még oly képzett és szakmailag felkészült is — már csak idejétmúlt zavaró tényező lesz az informatikai társadalom sokkal fejlettebb információközvetítési gépezetében. És akkor minden más megvilágításba kerül: ez az INFO a jövő előhírnöke volt.

Komáromi Nokia

A Nokia nemrégiben felavatott komáromi gyárában a hordozható telefonok és egyéb mobil kommunikációs eszközök gyártása mellett kutatási és fejlesztési tevékenységet is folytatnak. A 25 milliárd forintos, zöldmezős beruházás termelésének jelentős része a földrajzilag hozzánk közel eső országokba irányul majd. Az 500 fős induló létszámot fokozatosan mintegy 1300 főre kívánják emelni.

A Telnet gyarapodása

Májusban jelentős külső tőkebevonással részvénytársasággá alakult a Telnet Kft. A tőkeemelés mértéke 547 millió Ft. A három befektető közül a legnagyobb tulajdonhányadot (77%-ot) a CEIIF (Central European Internet Investment Fund) szerezte meg. A washingtoni székhelyű vállalkozásnak ez az első nagy befektetése ebben a térségben. A másik tulajdonos az elsősorban hazai magánbefektetőkből álló, kockázati befektetéssel foglalkozó Eclipse Rt, a harmadik a Texasban bejegyzett Danubius Net Investors (DNI).

Microsoft — Sun — AC

Az Andersen Consulting (AC) május elején bejelentette, hogy eProcurement néven az elektronikus kereskedelem területén működő új céget hozott létre, amely a Sun és az iPlanet E-Commerce Solutions (Sun/Netscape Alliance) technológiáival lesz felszerelve. A Sun beugróként hozzájárul az erőforrások előteremtéséhez, az új cég pedig 300 millió dollár értékben vásárol a Sun-tól hardvert és szolgáltatásokat. A megállapodást az teszi külön érdekessé, hogy

előzőleg az Andersen Consulting a Microsofttal is kötött egy szerződést az Avanade nevű közös vállalat létrehozásásáról, szintén az elektronikus kereskedelem erősítését tűzve ki célul. Ezzel a két szerződéssel az AC „virtuális közvetítőként" szerepelhet az egymással perben álló két nagyvállalat között.

Füstbe ment házasság

A szakma nagy érdeklődéssel várta a Corel és az Inprise/Borland 2000. február 7-én bejelentett összeolvadásának eredményeit, hiszen annak deklarált fő célja a linuxos fejlesztés volt. Megjelent ugyan a Corel Linux új verziója és egy linuxos irodai programcsomag, másik oldalon pedig a Borland Java Builder fejlesztőeszköznek a Linuxot is támogató változata, de az összeolvadási folyamat elé egyre több akadály gördült. A befektetők lelkesedése gyorsan elpárolgott, a Corel az első negyedévet a vártnál nagyobb veszteséggel zárta, részvényárfolyama rohamosan csökkent, április végére pedig készpénztartaléka is veszélyes mélypontra süllyedt.

Gyorsan kellett cselekedni, felülvizsgálva a házasságot. Mivel az Inprise részvényeseit is sújtotta, hogy részvényárfolyamukat a Corel magával rántotta, május elejére kétségessé vált a tervezett összeolvadás kivitelezhetősége. Május 16-án végül bejelentették, hogy „kölcsönös megegyezéssel" elállnak a két cég tervezett összeolvasztásától. A szakítás a kényszerítő körülmények hatására, barátságosan történt, és a további együttműködés előtt nyitva hagyták a kaput. A szerződésbontás hírére mindenesetre gyorsan emelkedett mindkét cég részvényeinek árfolyama.

Hátsó ajtó — becsukva

Az internet biztonságával foglalkozó egyik csoport hátsó bejáratot fedezett fel a Red Hat-ben, a jelenleg egyik legnépszerűbb Linux disztribúcióban. Az Internet Security Systems véleménye szerint emiatt illetéktelenek is elérhettek és módosíthattak fájlokat a rendszerben. A hiba a webmesterek munkáját könnyítő „Piranha" alkalmazási csomagban volt, és elsősorban a webszolgáltatási célokra telepített gépeket érintette. A hiba felfedezése után néhány nappal, április 24-én az ftp://updates.redhat.com/6.2 címen megjelentek a „Piranha" javított csomagjai, és azokat CD-mellékletünkre is feltettük.

Fókusz

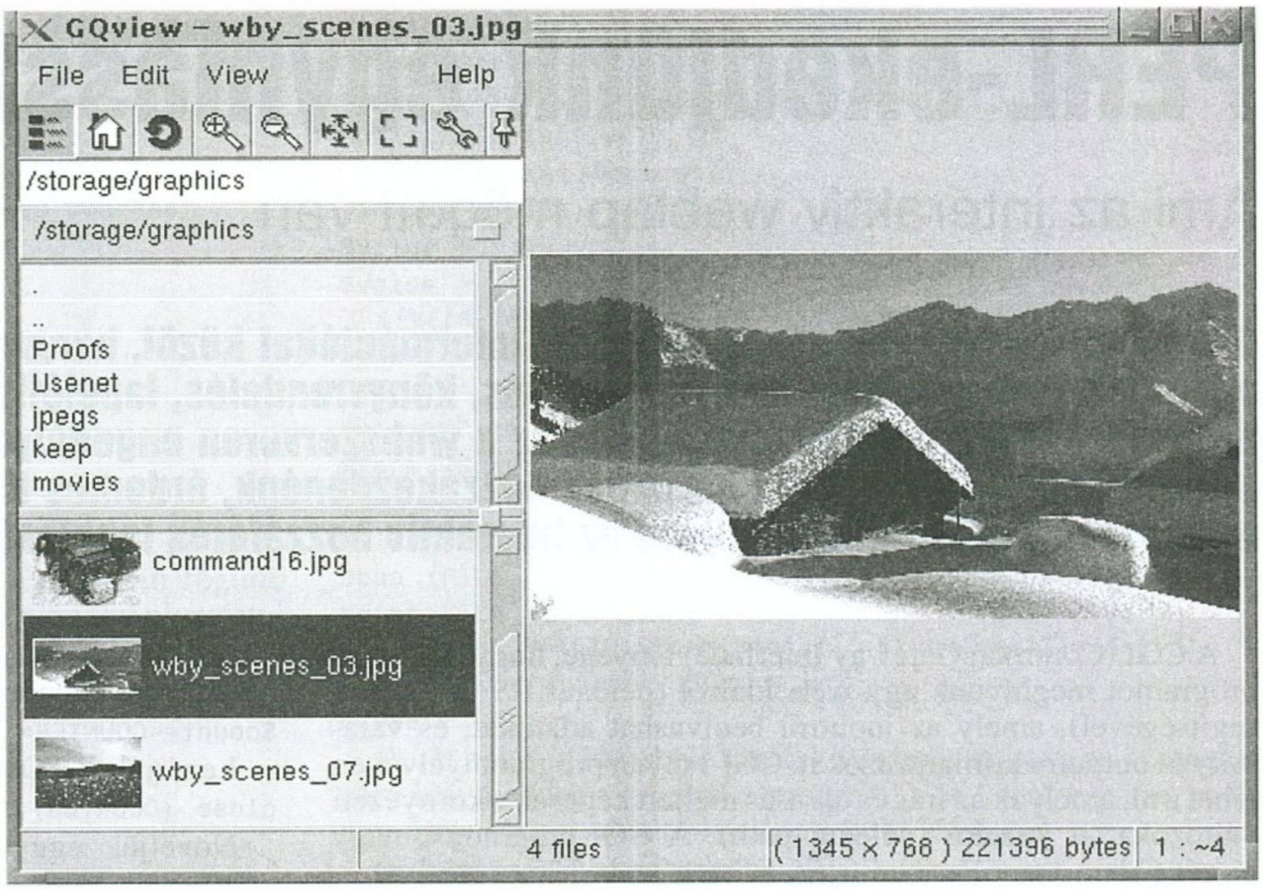
Nem kell sokat bizonygatnunk a kompatibilitás jelentőségét az egyre heterogénabbá váló számítástechnikai környezetben. A hónap témájához kapcsolódva a CD-melléklet Fókuszába is az ehhez kapcsolódó anyagokat helyeztük.

A programozás iránt érdeklődőknek először a Ralf Brown által készített megszakítási listát emeljük ki, amelynek segítségével a Microsoft és más gyártók DOS-verzióinak belső függvényeit lehet kezelni. Számos belső funkció gyors elérésében és kihasználásában segít, és a többi platform DOS-emulátoraihoz is hasznos lehet. Assembler vagy egyéb olyan fejlesztőkészlet szükséges hozzá, melyben Assembly nyelven írt blokkokat tudunk beépíteni, vagy tárgykódú fájlokat (*.OBJ) lehet lefordítani. Hogy mennyire népszerű ez a lista, azt igazolják a használatát segítő, részben más programozók által készített segédeszközök, melyekből szintén találhatnak néhányat CD-mellékletünkön.

A programozókat segíti egy másik csomag is, a Linuxra készült DOSEMU legfrissebb verziója, amely alapvetően a FreeDOS-sal kompatibilis DOS-emuláció mind teljesebb megvalósítását célozza. RPM és TGZ formátumban egyaránt közreadjuk, az utóbbiban a forráskódokat is megtalálják. A Microsoft rendszerekkel kapcsolatot tartó másik Linux eszköz, a Wine révén Linuxon is futtathatók a Windowsra írt alkalmazások. (Megfelelő paraméterezéssel még a nem implementált DLL-eket használó programok is.)

Szintén windowsos programoknak, illetve a Windows környezetnek a portolását (átírását) szolgálja egy másik gyűjtemény, a Willows is. A forráskóddal együtt szabadon használható könyvtár anyagát a „GNU Library General Public License” figyelmebevételével lehet használni, módosítani. Részletes leírása is megtalálható a mellékelt dokumentációban, illetve további információk olvashatók róla a http://www.willows.com címen.

A Windows 2000 használóinak közreadunk egy kompatibilitási listát, sajnos csak az 1999. év végi állapotnak megfelelően, mert offline formában nem sikerült újabbat felkutatni. Ennek ellenére hasznos támpont lehet, a folyamatosan frissített adatok pedig elérhetők a http://www.microsoft.com/hcl/default.asp online keresőrendszerében.



Az egy PC-n több virtuális gépet megvalósító VMware 2.0 a 10. oldalon kezdődő cikkből részletesen megismerhető. Maga a program is rajta van CD-mellékletünkön, a telepítéséhez szükséges regisztráció pedig a http://www.vmware.com címen bonyolítható le.

Vendégoldal

A CD Vendégoldal rovatában közreadott anyagok a programozáshoz és a napi munkakörnyezet karbantartásához egyaránt hasznosak. Az előbbi kategóriába tartozik például a NetRexx és Java alapú programozást segítő frissítés az IBM és a Sun kínálatából, valamint Kádár Zsolt folytatásokban közölt NetRexx tanfolyama.

A Microsoft anyagai jelentős arányban képviseltetik magukat ebben a rovatban. A Windows 2000 finomhangolásáról szóló angol nyelvű tanulmány közvetlenül olvasható CD-nkről Word 97 dokumentum formájában. A platformfüggetlen RTF-be konvertált formát tömörítettük, mert a bőséges képanyag okán mérete meghaladja a 30 MB-ot, de elkerülhető a kibontása, ha megnyitásához a ZipMagic-et használják, amely a Fészerben megtalálható.)

A Microsoft kiadta az Office 2000 szervizcsomagját, és szintén az Office 2000-hez kapcsolódik Front Page 2000 SDK, valamint a teljes csomag biztonsági frissítése. Ugyancsak megtalálhatók a korábbi Outlook verziók attachment kezelését javító csomagok.

Egyszerű képnézegetőként indult, idővel egész csomaggá terbélyesedett az ACDSee, amely már a 16 bites Windows világában is igen népszerű volt. A 3-as verziótól kezdve megváltozott koncepcióval hozzák forgalomba: a regisztrálható shareware programból kereskedelmi termék lett. A sokoldalú szolgáltatásokat bemutató demót érdemes kipróbálni, bővíthetőségéből pedig izelítőt ad a CD-mellékletre ugyancsak felkerült plug-in gyűjtemény, melyek a különböző képformátumok megjelenítésére szolgálnak. Az ACDSee jól illusztrálja, hogy a kompatibilitás megoldható ilyen szabad bővítések készítésének támogatásával is.

Frissítések

A Szerszámosládából érdemes megemlíteni a WinZip 8 végleges változatát, mely a 32 bites Windows platformokon egyike a legteljesebb szolgáltatást nyújtó és könnyen kezelhető shareware archívumkezelőknek.

A Loveletter féreg és a belőle készült mutációk robbanásszerű terjedése hirtelen a figyelem előterébe tolta a vírusok elleni védekezést. Erről részletesen is írunk a 48. oldalon, a mellékletre pedig feltettük néhány antivírus program „féregirtó” frissítéseit, melyek CD-zártáig megjelentek.

Simay Endre István

wxPython

PERL-ekedjünk egymással

Ami az interaktív weblap mögött van

A weboldalak jelentős része nemcsak információkat közöl, hanem adatokat is kér a látogatóktól: vendégkönyv, feliratkozás, könyvrendelés, lapelőfizetés stb. Ha ilyeneket akarunk elhelyezni, ahhoz a webszerveren engedélyeztetni kell a CGI futtatását, de mielőtt bármihez hozzákezdenénk, érdemes tisztázni a rendszergazdával, hogy milyen az interaktív hozzáférés lehetősége.

A CGI (Common Gateway Interface) lényege, hogy egy adott programot meghívunk egy weboldalról (például IIS parancsok segítségével), amely az inputról beolvashat adatokat, és valamilyen outputra kiírhatja azokat. CGI-t olyan programnyelveken lehet írni, amelyek az írás és olvasás mellett képesek a környezeti változókat is kezelni (például path). A CGI-programok nagy részét C-ben vagy Perlben írják, főleg a Unix/Linux rendszereken. A C mellett szól gyorsasága, a Perl mellett pedig az, hogy újrafordítás nélküi bármely rendszeren elindítható, ahol van Perl-futtató.

A szerveren futó CGI-program kimenete egy leegyszerűsített HTTP válaszként jelenik meg, amelyet a böngészőprogram értelmez. A HTTP fejlécében jelezzük, hogy milyen típusú adatot fogunk küldeni a böngészőprogramnak. Például, ha úgy kezdjük a kiírást, hogy Content-type: text/html, majd egy üres sort hagyunk (két soremeléssel), akkor már egyértelmű, hogy szimpla HTML oldal következik, és ki is írhatjuk a HTML-kódokat:

```
<HTML><TITLE>Próba</TITLE><BODY>Itt egy egyszerű
próba</BODY></HTML>
```

Fontos tudni, hogy a CGI-ket „távoli felhasználóként" futtatjuk, így a programok és az általuk használt fájlok hozzáférési jogait az ownernek kell megadnunk, mint ahogyan weboldalainknak is a Read és eXecution jogokat: chmod 755 (user=mindenki, group=olvasás+futtatás, owner=olvasás+futtatás /láthatóság/).

Példáinkban a Perl nyelvet használjuk, futtatási lehetősége minden rendszeren megvan (DOS, Windows, Unix stb.).

Nézzünk meg egy kezdetleges számlálót, amely meghívásakor beolvas egy counter.txt állományban szereplő számértéket (számlálónk aktuális állását). Ezt növeli, majd visszaírja a counter.txt-be, és egyben kiírja a képernyőre is az értékét.

Hozzunk létre egy counter.txt nevű állományt, és írjunk bele egy nullát (0). Ez számlálónk kezdőértéke. Adjunk neki jogokat. Ne feledjük, hogy az owner matatni fogja az állományt, és nemcsak olvasási jogot kell neki adni, hanem írási jogot is, hiszen ugyanebbe az állományba írja vissza a számláló mindenkori állapotát:

```
~/public_html$ cat > counter.txt
0
```

(Lezárjuk <Ctrl+C>-vel.)

```
~/public_html$ chmod 766 counter.txt
```

Ezekután készítsük el a programot, mondjuk counter.cgi néven. Megadjuk a CGI-futtatónak a programnyelv futtatójának/értelmezőjének teljes útvonalát:

```
#!/usr/bin/perl
```

Megjegyzésként írjuk oda azt is, hogy mi ez a program:

```
# Egyszerű kis számlálóprogram a weboldalamhoz.
```

Egy változóban megadjuk a számláló értékét tartalmazó fájl nevét:

```
$counterfile="counter.txt";
```

Megnyitjuk a counter.txt állományt:

```
open (COUNTER, "<$counterfile");
```

Beolvassuk a count változóba a COUNTER fájl (azaz a megnyitott counter.txt) egy sorát (jelen esetben egészét):

```
$count=<COUNTER>
```

Lezárjuk a COUNTER állományt:

```
close (COUNTER);
```

Növeljük eggyel a számláló értékét (ami most a $count változóban van):

```
$count++;
```

(Ez ekvivalens a $count=$count+1; paranccsal)

Megnyitjuk írásra (létrehozzuk újból) a counter.txt-t (amelynek nevét még mindig a $counterfile változó tartalmazza:

```
open (COUNTER, ">$counterfile");
```

Beleírjuk a már megnövelt számlálóértéket:

```
print COUNTER "$count";
```

Aztán újból lezárjuk a fájlt:

```
close COUNTER;
```

Kiírjuk a böngésző számára, hogy HTML kódot küldünk, és két soremeléssel lezárjuk a böngészőnek szóló fejlécet (az \n karakterpáros a C-hez hasonlóan egy soremelést jelent):

```
print "Content-type: text/html\n\n";
```

Majd HTML-címke (tag) nélkül egyszerűen kiírjuk számlálónk értékét:

```
print "$count";
exit (0);
```

Miután megadtuk a counter.cgi-nek a (chmod 755 counter.cgi) jogokat, már csak az index.html-be kell beírni — bárhová — a következő sort:

```
<!--# exec cgi="counter.cgi" -->
```

Az oldalon ott fog megjelenni számlálónk értéke, ahová elhelyeztük az adott hívást. Például:

```
Te vagy a(z) <!--# exec cgi="counter.cgi" -->.
látogatónk!
```

Eddig még semmi érdekes nincs a dologban, hiszen olvastunk egy fájlból, írtunk is oda, valamint a képernyőre is küldtünk valamit. Mi a helyzet akkor, ha mondjuk kíváncsiak vagyunk arra, hogy a látogatók mely linkjeinkre hányszor kattintanak rá. Ez is megoldható. A dolog lényege az, hogy meghívunk egy programot a link címével, amely program regisztrálja a rákattintás tényét, és továbbadja a megjelentetést a linkünknek. Honlapunkon a következő módon hivatkozhatunk a st.cgi (statisztika) programra:

```
<A HREF= "st.cgi?url=
http://www.gepnev.hu/alkonyvtar/linkfile.kiterjesztes">
Kattints ide!</A>
```

A lényeg az, hogy a programnak átadunk egy url nevű változót, amelynek értéke az a link, amelyre eredetileg (is) mutatna a „kattints ide" szöveg. Érdekes és gyakran használt formula a cgi-kben az adott paraméterek lekérdezése. Lássuk az st.cgi programot:

```
#!/usr/bin/perl
use CGI;
$query = new CGI;
$Url= $query->param("url");
```

```
($sec, $min, $hour, $day, $month, $year, $wday, $yday,
$isdst)=localtime;
$remote_address = $ENV{'REMOTE_ADDR'};
split (/\./, $remote_address);
$packed_address = pack ("C4", @subnet_numbers);
($remote_host) = lc(gethostbyaddr ($packed_address,
2)); $month++;
$statfile="latogato.txt";
open (F,">$statfile");
print F „$url $remote_host $hour:$min:$sec\n";
close F;
print "Content-type: text/html\n";
print "Location: $Url\n\n";
exit (0);
```

A paraméterek beolvasására van egy bonyolultabb, de megbízhatóbb módszer is, amelyet következő példánkban fogunk alkalmazni. Az előző szkriptbe is be lehetett volna így olvastatni a programnak átadott paramétereket:

```
read(STDIN,$buffer,$ENV{CONTENT_LENGTH});
split(/&/, $buffer);
foreach $pair (@pairs){
($name, $value) = split(/=/, $pair);
$value =~ tr/+/ /;
$value =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C",
hex($1))/eg;
$value =~ s/~!/ ~!/g;
$value=~ tr/;//;
$value=~ tr/=//;
$form {$name} = $value;
}
```

A fenti példa egy HTML oldaltól megkapott form tartalmát szedi szét a $form{mezoneve} formában. Ha az előző példánkban szereplő url változót szeretnénk megkapni, tehát nem formtartalmat olvasunk be, hanem az IIS parancsnak megfelelően futtatunk, akkor a

```
read(STDIN,$buffer,$ENV{CONTENT_LENGTH});
```

sort a következőre kell cserélni:

```
$buffer= $ENV{'QUERY_STRING'};
```

A végére nézzünk példaként egy rövid kis vendégkönyvet, amelyet több részre bontva állítunk össze.

Először is kell egy sima HTML oldal, ahová be lehet gépelni az üzenetet.

```
(beir.htm):
```

(Csak az elengedhetetlen HTML elemeket közöljük, a többit mindenki maga formázhatja meg.)

```
<HTML>
<HEAD>
<TITLE>Itt üzenhetsz!</TITLE>
</HEAD>
<BODY>
...
<FORM ACTION="vendegk.cgi" METHOD=POST>
Neved:
<input type=text name=nev size=50>
E-mail címed:
<input type=text name=email size=30 maxlength=60>
Üzeneted:
<TEXTAREA NAME=uzenet COLS=60 ROWS=5></TEXTAREA><BR>
<input type=submit value="Elküld">
<input type=reset value="mégsem">
</FORM>
</BODY/HTML>
```

A <FORM action=...> a hivatkozás helye egy vendegk.cgi nevű programra, amelyet meg kell írnunk; ez kezeli a form tartalmát. Hozzunk létre egy üres uzenetek.htm fájlt

```
cat >uzenetek.htm
```

amelybe a program bejegyzi az érkezett üzenetek szövegét HTML kódokkal tarkítva, majd magát a vendegk.cgi-t is írjuk meg:

```
#!/usr/bin/perl
# Paraméterek beolvasása
read(STDIN,$buffer,$ENV{CONTENT_LENGTH});
split(/&/, $buffer);
foreach $pair (@pairs){
($name, $value) = split(/=/, $pair);
$value =~ tr/+/ /;
$value =
~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
$value =~ s/~!/ ~!/g;
$value=~ tr/;//;
$value=~ tr/=//;
$form {$name} = $value;
}
# Kiírjuk fájlba (uzenetek.htm) a form tartalmát
open (FILE, ">>uzenetek.htm") || die;
print FILE "$form{nev}
(<A HREF=\"MAILTO:$form{email}\">$form{email}</A>)\n";
print FILE "$uzenet HR\n";
close FILE;
```

A két belső idézőjel elé \ kell, különben azokat a print utasítás záró idézőjeleként kezelné a program, a \n karakterpáros pedig egy újsor (kocsi vissza) karaktert küld kiíráskor.

```
# Megköszönjük a hirdetést
print "Content-type: text/html\n\n";
```

A két újsor karakter fontos, különben a böngésző nem tudja jól értelmezni a most kiküldött kódot; ezzel jelezzük a webszervernek, hogy a fejléc után maga a kód jön.

```
print <<SZOVEG_VEGE
<HTML>
<HEAD>
<TITLE>Köszönjük üzenetét</TITLE>
</HEAD>
<BODY>
...
Köszönjük üzenetét, amely a következőképpen került
regisztrálásra:
Neve: $form{nev}
E-mail címe: $form{email}
Üzenete: $form{uzenet}
...
</BODY></HTML>
SZOVEG_VEGE;
```

Mindaddig írja folyamatosan a szöveget, amíg a print utasításban (print <<) szereplő kifejezés fel nem bukkan (jelen esetben SZOVEG_VEGE).

A végén még küldünk magunknak egy e-mailt, hogy új bejegyzés jött a vendégkönyvbe:

```
$sajatcim='sajat@email.cim';
open (MAIL, "|/usr/lib/sendmail $sajatcim");
print MAIL "To: $sajatcim\n";
print MAIL "Subject: Uj bejegyzes a vendegkonyvbe\n\n";
print MAIL <<SZOVEG_VEGE
Uj bejegyzes erkezett a vendegkonyvbe a kovetkezo
adatokkal:
Neve: $form{nev}
E-mail cime: $form{email}
Uzenete: $form{uzenet}
SZOVEG_VEGE
print MAIL "\n.\n";
```

Jelezzük a sendmail-nek, hogy ez a levél vége:

```
close MAIL;
exit (0);
```

Ez már csak felesleges plusz, a program vége...

Ezek után még egy dolgunk van, hogy egy oldalba beépítsük az eddigi üzeneteket, mondjuk a begépelő oldal aljára (beir.htm — lásd fentebb):

```
<!--# include file="uzenetek.htm" -->
```

Pál Zsolt
pille@elender.hu

Szelídítsünk pingvint — I.

A Red Hat Linux 6.1 installálás utáni konfigurálása

A legsikeresebb alternatív operációs rendszert, a Linuxot gyakran éri szemrehányás, hogy telepítése nehézkes, időt rabló, kockázatos, ezért csak türelmes profiknak és megszállott amatőröknek javasolható. Néhány éve, amikor jobbára még csak a Slackware disztribúcióval próbálkozhattunk, ez az állítás talán igaz is volt. Manapság azonban a Red Hat, a Mandrake és a többi disztribúció is sokkal barátságosabb önálló installálókészlettel rendelkezik, ezért jóval többen megbirkóznak a telepítés feladatával.

Bármennyire is automatizálható a telepítés, meggyőződésem, hogy egy operációs rendszer installálásához szükség van a leendő felhasználó aktív részvételére. Ez szemben áll azzal a felfogással, amely szerint az a hardver automatikus feltérképezése után minden állományt és programot (tudtunk és akaratunk nélkül) fel kell másolni a merevlemezre, függetlenül attól, hogy szükség van-e rá, vagy sem. Én ezt a módszert inkább tartom a felhasználó félrevezetésének, semmint felhasználóbarát magatartásnak.

Ne sajnáljuk az időt!

Egy átgondolt, megtervezett oprendszer-installálás hosszabb távon sokszorosan megtéríti a ráfordított időt és energiát, még akkor is, ha nem elsőre, hanem csak harmadjára vagy negyedjére sikerül. Ilyenkor gondoljunk mindig arra, hogy minden elrontott installálás járhat olyan tanulságokkal is, amelyeknek később még hasznát vehetjük...

A számítástechnikai lapokban arról azonban már kevés szó esik, hogy az installálás csak az operációs rendszer munkába állításának első lépése. Ezt egy hosszabb, aprólékos munkával kitöltött fázisnak kellene követnie, amit a következőkben installálás utáni konfigurálásnak fogunk nevezni. Persze nehéz ezt pontosan definiálni, hiszen az installálás és konfigurálás akár egységes egészként is felfogható, és egy-egy műveletről nehéz is eldönteni, hogy igazából melyik fázishoz tartozik. Én mégis azt remélem, hogy miután végigtekintettük egy konfigurálás részleteit, érthetőbb lesz, miért tartom jobbnak, ha két szakaszból állónak tekintjük az egészet.

Biztonságos boot

A konfigurálást nekünk is ott érdemes elkezdenünk, ahonnan az operációs rendszer indul: a betöltődés, a boot folyamatától. Elsősorban most mégsem a partíciós tábla és a bootszektor szerepéről szeretnénk szólni, hanem az ezzel kapcsolatos biztonsági résekről. Ha ezekről megfeledkezünk, hiába telepítjük Linux rendszerünket a legnagyobb gondossággal, hiába választunk a root felhasználónak nehezen feltörhető jelszót. Gondoskodnunk kell a boot folyamat védelméről is, különben oda a biztonság.

A Linux betöltését a lilo nevű program végzi. Ez többek között kiírja saját promptját, majd vár a felhasználó által megadott paraméterekre. Paraméter lehet a betöltendő operációs rendszer megnevezése, de megadhatunk plusz paramétereket is a Linux kernelnek. Az egyik ilyen paraméter a single. Ezt rendszerkarbantartásra használhatjuk, ekkor ugyanis a Linux egyfelhasználós módban indul el. Ez még nem is lenne baj, a gond ott kezdődik, hogy ilyenkor a Linux nem kér tőlünk jelszót, hanem automatikusan belépteti a root felhasználót, akinek ilyenkor módja van a jelszófájl manipulálására, így elérheti, hogy a Linux normál indításkor sem fogja kérni a root jelszót.

Hogy ezt megelőzzük, a lilo által használt konfigurációs fájlt kell korrigálnunk. Ez a fájl az /etc/lilo.conf. Ide tehetünk egy olyan jelszót, melyet single módban is kérni fog a Linux. Ehhez a password=<jelszó> sort kell a fájlban elhelyeznünk. Mivel azonban ez a jelszó innen kiolvasható, a védelmi kódot úgy kell átállítanunk, hogy a fájlt csak a tulajdonosa olvashassa. Ez elvégezhető a chmod 600 /etc/lilo.conf paranccsal. Hogy a Linux normál indításkor ne kérje ezt az extra jelszót (ekkor ugyanis természetesen a root amúgy sem léphet be jelszó nélkül), írjuk be a restricted kulcsszót is a megfelelő helyre.

Még egy előnye van ennek a plusz jelszónak. Ha például DOS-t is indítunk a lilo-val, és a password kulcsszót globális érvényességgel használjuk (mint ahogy a példában látható), akkor a DOS indítása sem lehetséges ennek a jelszónak az ismerete nélkül. Így a lilo DOS partíciónk védelméről is gondoskodik.

Egy, az imént elmondottaknak megfelelő lilo.conf így néz ki (a *-ok helyére természetesen saját jelszónkat kell beírni):

```
boot=/dev/hda
map=/boot/map
install=/boot/boot.b
prompt
default=dos
password=*******
image=/boot/vmlinuz-2.2.12-20
            label=linux
            initrd=/boot/initrd-2.2.12-20.img
            read-only
            root=/dev/hda5
            restricted
other=/dev/hda1
            label=dos
```

Ne felejtsük el a módosítás után a lilo parancsot kiadni, mert a konfigurációs fájlban történt változtatások csak ennek lefuttatásával válnak érvényessé.

A munkakörnyezet kialakítása

A következő feladat a kényelmes munkakörnyezet kialakítása lesz. Mit is értünk ezen? A Linux előtt a legtöbben használtak más operációs rendszert. Ha nem Unixot, hanem mondjuk DOS-t vagy VMS-t, akkor sok olyan körülményt megszokott, amelyről nem szívesen mondana le. Parancsok, munkamódszerek, batch eljárások, színek, gyorsbillentyűk és egyebek. Érdemes ezek közül lehetőleg minél többet megtartani Linux alatt is. Első teendőnk a DOS Config.sys és Autoexec.bat párosának megfelelő állományok beállítása. A Linux alatt nincs ezeknek egy az egyben megfelelő állománypáros, funkcionális hasonlóság azonban van.

Az állományok módosításához mindenképpen szükségünk lesz egy jó és egyszerű szövegszerkesztőre. A legjobb talán a Midnight Commander beépített editora. Ha elindítottuk a prog-

ramot, érdemes néhány opcióját is azonnal beállítani. Ilyen a backup és rejtett fájlok megjelenítése, a legördülő menü, a megerősítő kérdés nélküli kilépés és mentés, a Lynx típusú navigálás a könyvtárak között.

A Midnight Commanderrel mindenképpen érdemes megismerkedni, mielőtt a komolyabb rendszerkonfigurálásba belekezdenénk. Az MC megjelenésében, működésében és használati módjában is a DOS alatt rendkívül népszerű Norton Commander, illetve a későbbi Volkov Commander méltó rokona, s nélküle egy DOS-ról Linuxra váltó felhasználó kissé elveszettnek és tanácstalannak érezné magát.

A Midnight Commander nemcsak a fájl- és könyvtárkezeléssel kapcsolatos Linux parancsok megtanulása alól mentesít bennünket — éppen a kritikus ismerkedési periódusban —, hanem a vi szövegszerkesztő rejtelmeiben való elmélyüléstől is. Ezenkívül olyan extra szolgáltatásokat nyújt, mint az archívumok könyvtárként való kezelése. Ezen azt kell érteni, hogy a tar által készített (és esetleg gzip-pel is tömörített) archívumokat a hagyományos könyvtárakkal analóg módon használhatjuk, amennyiben nem rendelkeznek futtatható attribútummal, ami egy DOS/Windows partícióról másolt fájlra automatikusan rákerül, tehát ezt az attribútumot előbb le kell szedni, például a chmod-dal.

Még az archívumok létrehozását (fájlok és az egész könyvtárstruktúra tömörítését, archívumba helyezését) is megoldhatjuk az F2 billentyűre megjelenő felhasználói menü segítségével. És ami még hasznosabb, a Red Hat csomagkezelőjének rpm típusú állományaiba is belenézhetünk, ami különösen hasznos, ha például egy csomag telepítése után nem találjuk, hogy hová kerültek a végrehajtható állományok, vagy található a dokumentáció. Nagyon jó lehetőség az is, hogy általa az ftp szerverek könyvtárait ugyanúgy elérhetjük, mint a lokális könyvtárakat. Ez a későbbi csomagtelepítéseket vagy a javítókészletek telepítését könnyítheti meg.

Fel kell hívnunk azonban a figyelmet egy veszélyre is. A Midnight Commanderrel az állományok törlése ugyanolyan könnyű, mint a DOS alatti fájlkezelőkkel. Csakhogy a törölt állományok Linux alatt nem állíthatók vissza. (Vannak ugyan bonyolult módszerek és olyan fájlrendszerek, amelyekkel a visszaállítás megoldható, de egyelőre fogadjuk el, hogy a törölt állományok végleg elvesznek.) Ha tehát olyan állományt akarunk törölni, amelyről nem tudjuk pontosan, hogy mire való, inkább helyezzük át azt egy átmeneti tárolóhelyre, vagy nevezzük át. Előfordulhat ugyanis, hogy valami a legközelebbi indításkor nem indul el amiatt, hogy egy ártatlannak látszó konfigurációs vagy ideiglenes állományt töröltünk.

Háttérprogramok

A Linux operációs rendszer futtatása során több háttérprogramot, ún. démont indít el (hasonló feladatot látott el, mint a DOS-ban az Autoexec.bat, amikor például lemez cache programot indítottunk el belőle). Érdemes ezeket a háttérprogramokat számba venni, és csak azok indítását kérni, amelyekre tényleg szükségünk van. Értékes memóriát és időt takaríthatunk meg az átgondolt szelektálással. Szerencsére a Red Hat Linux egészen egyszerű módot kínál a démonok indításának szabályozására, az ntsysv parancsot. Konzolról (parancssori ablakból) elindítva azt, a szükséges készlet listából összeválogatható. Egy ilyen összeállítást mutatunk be az alábbiakban:

atd: az at paranccsal végzett programindításhoz;
crond: ütemezett programindításhoz;
gpm: segédprogram az egérrel történő kivágási és beszúrási műveletekhez virtuális konzolon;
inet: internet démon;
lpd: démon a hálózati nyomtatáshoz;
network: hálózati démon;
portmap: démon a Windows NT és a Windows 9x kiszolgálók lemezeinek és nyomtatóinak eléréséhez;
xfs: fontszerver az X Windowhoz.

A munkakörnyezethez tartozik a környezeti változók és a parancshelyettesítők mellett az /etc/bashrc állomány is, a DOS Autoexec.bat távoli rokona. Ide írhatjuk be azokat a rendszerszintű parancsokat, amelyeknek minden bejelentkezéskor le kell futniuk. Nézzünk egy példát:

```
# /etc/bashrc
PS1="\[\033[0;32m\]\u@\h \w>\[\033[0m\]"
alias bye=logout
alias del=rm
alias dir='ls -laF --color'
alias setx=chmod +x
alias x=startx
PATH="$PATH:/main/bin:."
```

Az első sor megjegyzés, a következő viszont a PS1 környezeti változót állítja, meghatározva, hogy miképpen nézzen ki a shell készenléti jele, azaz a prompt. Jelen esetben a parancsra várakozva ez a felhasználó nevét, a '@' jelet, a gazdaszámítógép (host) nevét, majd szóközzel elválasztva az aktuális directoryt, végül a '>' karaktert fogja megjeleníteni zöld karakterekkel. Érdemes megjegyezni, hogy Linux platformon a prompt beállításának igen sok variációja lehetséges, akár programokat futtathatunk, és azok eredményét bevihetjük a készenléti jelbe. A fent bemutatott beállítás az egyszerűbbek közül való, de még ez is tartalmazza az eligazodáshoz szükséges alapinformációkat.

A prompt beállításakor érdemes figyelni arra, hogy minden olyan információt beletegyünk, ami ahhoz kell, hogy gyorsan azonosítani tudjuk, hol is vagyunk éppen. Így nem fordulhat elő, hogy egy másik számítógép lemezéről törlünk valamit, abban a tudatban, hogy még a saját gépünkön dolgozunk.

A következő néhány sorban parancshelyettesítőket definiálunk az alias paranccsal. Ennek több oka lehet: a már megszokott neveket szeretnénk továbbra is használni (del); gyakran használt hosszabb neveket szeretnénk rövidíteni (x); esetleg gyakori kapcsolókombinációt teszünk alapértelmezéssé (dir).

Az utolsó sor a végrehajtható állományok elérési útját állítja be. Itt semmi mást nem tettünk, mint kiegészítettük a rendszer által beállított útvonalat egy könyvtárral (/main/bin), amelyben majd az általunk készített shell szkriptek és futtatható programok lesznek (ezt a könyvtárat persze létre is kell hozni). Továbbá hozzáadtuk az útvonalhoz az aktuális munkakönyvtárat (.) is. Ez utóbbival kapcsolatban általában megoszlanak a vélemények. Rendszergazdák számára és olyan számítógépeken, amelyek külső hálózatról érkező támadásnak is ki lehetnek téve, biztonsági okokból nem ajánlják az aktuális könyvtár kapcsolását az elérési útvonalhoz. Egy otthoni, hálózat nélkül használt számítógépen azonban kockázat nélkül megtehetjük ezt.

DOS fájlrendszerek elérése

Akik egyelőre csak próbaképpen telepítik a Linuxot, természetesen megtartják a gépen a másik operációs rendszert is, például a DOS 6.22-es változatát. Minél gyakrabban használják a Linuxot, annál inkább erősödik az igény, hogy a DOS partíciókon lévő állományokat is elérhessék Linux alól, illetve, hogy oda-vissza mozgathassák a fájlokat a fájlrendszerek között.

A Linux erre is kínál megoldást. Nemcsak a DOS partíció érhető el, hanem számos más operációs rendszer eltérő típusú partícióit is hozzácsatolhatjuk a Linux fájlrendszeréhez (VFAT, HPFS, MINIX, ISO9660 stb.).

Példánkban két partíciót — ezek a DOS alól C: és D: meghajtóként érhetők el — kapcsolunk hozzá a Linux fájlrendszerhez, melynek egyik fontos jellemzője, hogy nincsenek meghajtónevek, hanem a külső fájlrendszerek meghajtóit is beleértve minden eszköz a gyökérkönyvtár (/), illetve az /mnt valamelyik alkönyvtáraként érhető el. Készítsünk tehát két könyvtárat /dos/c és /dos/d néven a két DOS meghajtó részére. (Bár a csatolásokat általában az /mnt alkönyvtáraihoz szokták elkészíteni, ez nem kötelező, a lényeg az áttekinthetőség.) A DOS meghajtók csatolását több módon is elvégezhetjük. Ideig-

lenesen a mount /dev/hda1 -t msdos /dos/c paranccsal tehetjük elérhetővé a C: drive-ot. Itt a /dev/hda1 az első IDE meghajtó első partícióját jelenti (amely DOS alatt a C:), a -t msdos a fájlrendszer típusa, a /dos/c pedig a már említett célkönyvtár.

Ha permanenssé szeretnénk tenni a hozzárendelést, erre is van több módszer. Az egyik a linuxconf program Access local drive opciója (a linuxconf KDE alól elérhető a Start menü RedHat szekciójában). De bármilyen módon is csatoljuk a meghajtókat, a változás az /etc/fstab fájlban jelenik meg, így ennek a fájlnak közvetlen módosításával is elérhetjük ugyanazt a végeredményt. Az egyszerűség kedvéért nézzük meg a végleges fájlt:

```
/dev/hda5    /            ext2     defaults           1 1
/dev/hda6    swap         swap     defaults           0 0
/dev/fd0     /mnt/floppy  ext2     noauto,owner       0 0
none         /proc        proc     defaults           0 0
none         /dev/pts     devpts   gid=5,mode=620     0 0
/dev/hda1    /dos/c       msdos    umask=177          0 0
/dev/hdc1    /dos/d       msdos    umask=177          0 0
```

Az utolsó két sorban láthatóak a változások, a többi még installáláskor került ide. Az umask paraméter magadja, hogy a DOS-os fájlok milyen elérési móddal jelenjenek meg a Linux alatt, jelen esetben ez írási és olvasási jogot ad a tulajdonos számára. A sor végén látható két szám a dump és az fsck programok számára szóló információ, a második 0 például azt jelenti, hogy a fájlrendszert nem kell a boot folyamata során ellenőrizni, így a betöltés lényegesen gyorsabb lesz.

A DOS meghajtókkal kapcsolatban érdemes szólni a szövegfájlok konvertálásáról is. A Linuxon (és a Unixon) a szövegfájlok rekordjait csak a soremelés karakter, DOS alatt a kocsivissza-soremelés karakterpáros választja el. Csatoláskor beállítható, hogy ez az oda-vissza konvertálás automatikusan megtörténjen-e. Ez még akkor sem ajánlható, ha csak a szövegfájlokra állítjuk be. A Linux ugyanis nem minden esetben tudja teljes biztonsággal eldönteni, hogy kell-e konvertálni vagy sem, így esetleg olyan esetekben is elvégzi az átalakítást, amikor nem kellene. Jobb, ha inkább magunk végezzük el a korrekciót. DOS alatt is számos olyan segédprogram létezik, amely erre alkalmas, sőt kialakulóban van egy olyan gyakorlat, hogy a szövegszerkesztők és a szövegkezelő segédprogramok egyaránt képesek mindkét rekordelválasztási módú szövegfájl feldolgozására, ami feleslegessé teszi a konverziót.

Fájl- és szövegkeresés

Gyakori feladat olyan fájlok megkeresése, melyekben egy adott szöveg előfordul. Erre a célra szolgál DOS alatt a Norton Utilities csomag ff fájlkereső és ts szövegkereső programja. Érdemes valami ehhez hasonló eszközt Linux alatt is elkészíteni.

Nézzük először a fájlkeresést. Ehhez a Linux find programját fogjuk felhasználni, csupán azért tesszük ezt egy shell szkriptbe, hogy egyszerűbb legyen a hívása. Készítsük el tehát az alábbi szövegfájlt ff néven, tegyük a /main/bin-be (lásd a PATH-nál leírtakat), tegyük végrehajthatóvá (setx ff), majd próbáljuk ki: ff / bashrc. Az eredménynek /etc/bashrc kell lennie.

Néhány szót a -mount paraméterről: ezzel elérhetjük, hogy a find ne keresgéljen a csatolt külső fájlrendszereken, amilyen az előbb tárgyalt /dos/c és /dos/d könyvtár. Ellenkező esetben ugyanis hosszabb keresésre kell felkészülnünk. Ha minden meghajtón keresni akarunk, akkor a Midnight Commanderben is találunk egy fájl- és szövegkereső menüpontot, ez is kiválóan alkalmas erre a célra.

```
#/bin/sh
#
#           ff : Find Files
#           Usage : ff <directory> <file>
#
find $1 -name $2 -mount
```

A szövegkereséshez gépeljük be az alábbi kis eljárást ts néven, tegyük ezt is végrehajthatóvá, majd próbáljuk ki. Szintén a find parancsot használja, amely a szövegkereséshez minden fájlhoz meghívja a grep programot, hogy az találat esetén kiírja az eredményt.

```
#/bin/sh
#
#           ts : Text Search
#           Usage : ts <directory> <text>
#
find $1 -mount -exec grep -l $2 {} \; 2> /dev/null
```

Ez utóbbi eljárás a teljes fastruktúrában történő kereséskor a /dev és a /proc könyvtárakban esetleg elakadhat, tehát ebben az irányban még tökéletesítendő.

Természetesen ezek igen egyszerű példák, céljuk inkább a szemléltetés, és a shell-programozás néhány vonásának bemutatása volt. Arra is szerettük volna felhívni a figyelmet, hogy a Linux alatti eszközök kezdetben esetleg idegennek, nehezen használhatónak, körülményesnek tűnnek, néhány egyszerű módszerrel lényegesen barátságosabbá lehet tenni őket.

Még két programot érdemes megemlíteni ebben a témakörben. A locate ugyanúgy fájlkeresésre használható, mint a find, de ez a megadott nevet névtöredékként értelmezi, ezért hosszabb lehet az eredménylistája.

A locate adatbázist használ a kereséshez, emiatt sokkal gyorsabb, de nem mindig naprakész. Első alkalommal, és minden olyan esetben, ha a fájlrendszer megváltozik, az updatedb paranccsal kell aktualizálnunk az adatbázist. Ezt az aktualizálást akár napi gyakorisággal automatikusan is elvégeztethetjük a cron ütemező segítségével.

A másik program a whatis, amely különösen a kezdeti időkben hasznos segédeszköz. Ez is adatbázist használ, melyet a makewhatis parancs hoz létre, és a megadott programokról egysoros leírást tud adni. Ez jól jöhet mondjuk a /bin könyvtárban történő felfedezőúton, amikor is a gyakran két-három betűs Linux parancsnevek szinte semmit nem árulnak el a program mibenlétéről.

Nyomtatóbeállítás

Induljunk ki most is egy egyszerű példából. Mondjuk van egy múzeumi darabnak számító Citizen 120-D nyomtatónk, mely még tökéletesen működik, és szeretnénk programlisták kinyomtatására használni. Erre tulajdonképpen alkalmasabb a tintasugaras nyomtatóknál.

A beállításhoz a printtool programot indítsuk el, s ehhez szükséges az X Window is. Elindíthatjuk a Vezérlőpult (Control Panel) alól, de van rá egyszerűbb mód: KDE alatt az Alt-F2 egy kis parancsablakot jelenít meg, amelybe beírhatjuk az X alatt indítani kívánt program nevét, jelen esetben: printtool.

Vegyünk fel két nyomtatót, az egyiket „text only" printerként (ez lesz az lp), a másikat, mivel nincs Citizen típus a nyomtatólistán, Epson 9 tűs nyomtatóként (ez az lp0 lesz — de a neveken persze változtathatunk). Ezzel a trükkel megoldható, hogy amikor egyszerű programlistát nyomtatunk, azt az lp-re küldjük, ami így sokkal gyorsabban, karakteres módban nyomtat, míg ha különböző fontokat, vagy grafikát tartalmazó dokumentumunk van, azt kinyomtathatjuk az lp0-ra küldve, grafikus üzemmódban.

Ha valakinek problémái lennének a nyomtató elérésével, az írja be az /etc/conf.modules fájlba a következő sort: alias parport_lowlevel parport_pc. Ez a sor ugyanis hiányzik a Red Hat 6.1 telepítésekor gépünkre kerülő modul betöltő állományából.

(Legközelebb a hálózati környezet és a grafikus felület beállításával foglalkozunk.)

Szűcs János
szucsj@josa.szabinet.hu

SGML nyelvek konverziója — II.

A fordításdefiníciós nyelv

Az előző részben eljutottunk odáig, hogy a fordítás során milyen jellegű feladatokat kell megoldani. A fordítás formális leírására létrehozott fordításdefiníciós nyelvben az ilyen jellegű feladatok megoldására alkalmas eszközöket, nyelvi elemeket kell biztosítanunk. A következőkben bemutatunk egy általunk kidolgozott nyelvet, amelyen a felmerülő feladatok jó része leírható, megoldható.

A fordításdefiníciós nyelv fordításdefiníciója ténylegesen szabályok sorozata, amelyet alkalmazni kell a lefordítandó dokumentum fordítása során. Egy szabály alapvetően 3 részből áll.

1. Az elsőben az adott elem fordítása előtti lépéseket kell összefoglalni, amelyek a kimenet egyetlen részét sem generálják, ám a helyes fordítás érdekében szükségünk van rájuk. Ezek általában a környezetfüggő elemek kezelésével kapcsolatos lépések. Ilyen például az UL és OL elemek előzőekben leírt fordítása során a list és a strat változó értékének beállítása.

2. Egy fordítási szabály második részében tulajdonképpen leírjuk, hogy a fordítás során keletkező absztrakt fában milyen elemek jelenjenek meg. Itt definiáljuk tehát a tényleges fordítást. A HTML H3 elemének bemutatott fordítását gyakorlatilag csak ebben a részben kell definiálnunk.

3. Egy szabály harmadik része összefoglalja, hogy milyen lépéseket kell megtennünk az elem lefordítása után. Ebben a részben is a környezetfüggő elemek kezelése a legjellemzőbb feladat, hasonlóan az első részhez. Példa lehet ennek a résznek a felhasználására a HTML LI elemének a fordítása, amikor a start változó értékét növeljük.

Kicsit programozói szemmel nézve: az első és a harmadik részben imperatív szerkezeteket használunk, azaz több egymást követő utasítást adhatunk ki; a középső részben viszont funkcionális módon írjuk le, hogy miből álljon az adott elem fordítása, azaz itt a definíció egy kifejezés.

Szabályok

A következőkben bemutatjuk a fordításdefiníciós nyelv néhány lehetőségét, és nagy vonalakban a nyelv alapfelépítését.

A fordításdefiníciós nyelvben egy szabály alapfelépítése az alábbiakban látható, ahol a pre és post szóval kezdődő rész a fordítás előtti és utáni tevékenységek leírására szolgál, a trans rész magát a fordítás leírását, míg az <elem_neve> kifejezés a kiindulási SGML nyelv egyik elemét tartalmazza:

```
define <elem_neve> as
    pre:  ..... /* Fordítás előtti feladatok */
    trans:   ..... /* A fordítás tényleges leírása */
    post: ..... /* Fordítás utáni feladatok */
```

Egy fordítási szabály első részében hozzáférhetünk a változók tartalmához, módosíthatjuk azokat, feltételes utasításokat használhatunk. Az alábbiakban a már említett UL és OL elemhez tartozó fordítási szabályokat mutatjuk be, HTML és WML közötti fordítás esetén:

```
define ul as
    pre:      set(list, "ul");
    trans:    trans(content);
define ol as
    pre:      set(list, "ol");
              set(start, "1");
    trans:    trans(content);
```

Mindkét definíció hasonló felépítésű. A lényegi rész a fordítás előtti változóbeállítás, amelyet a set utasítás segítségével tehetünk meg. Az utasításban meg kell adni a változó nevét, ahogyan majd hivatkozni fogunk rá, majd második argumentumként a változó értékét, amely mindig egy karaktersorozat. A beállított változóra a fordításdefiníció egyéb részeiben glob.<változó_neve> módon hivatkozhatunk (ahol a <változó_neve> egy konkrét változónevet takar, például list). Ez a hivatkozás bárhol előfordulhat a fordításleírásban, tehát nem csak az adott szabályban, ahol definiáltuk.

A fordítás előtti tevékenység leírása mellett itt definiáltuk az UL elem fordítását is, ami tartalmának lefordított változata. (Az UL elem tartalmát absztrakt szinten az UL elem leszármazottjai alkotják az absztrakt fában, konkrét karakteres formában pedig a nyitó és záró UL elem között előforduló elemek.) Ezt a definíciót a fordítási szabály középső részében adhatjuk meg a trans(content) kifejezés felhasználásával.

```
define li as
    trans:    if glob.list = "ol" then
                  glob.start + trans(content)
              else
                  "-" + trans(content);
    post:     if glob.list = "ol"
                  set(start, incr(glob.start));
```

A fentiekben fordítás utáni tevékenység definiálását láthattuk (ez az előző részben, a 6. ábrán bemutatott példafordítás leírásának egyik részlete). Ezek a tevékenységek teljesen azonos alakúak, mint a fordítás előtti tevékenységek. A példában egy új elem az incr(...) függvény használata, amely az argumentumában található karaktersorozatot mint számot értelmezi, és annak eggyel megnövelt értékét visszaadja karaktersorozat formában (tehát az incr("1") értéke: "2").

A szabály középső részében egy feltételes kifejezéssel találkozhatunk. Azt vizsgáljuk, hogy számozott vagy számo-

```
<html>
  <body>
    <h4>Példa</h4>
    <hr>
    <p>Hello</p>
    <p>Világ</p>
  </body>
</html>
```



7. ábra. Egyszerű HTML dokumentum és megjelenítése

zatlan listának az elemét fordítjuk-e. Ehhez megvizsgáljuk, hogy a list nevű globális változóban az ol karaktersorozat van-e (ezt a változót az OL, illetve UL elemek fordítása során állítjuk be). Ha a lista számozatlan, akkor egyszerűen egy '-' karaktert generálunk az LI elem tartalmának lefordított változata elé a WML dokumentumban. Ha számozott listával van dolgunk, akkor kiírjuk a start változó tartalmát, majd azt követi az LI elem tartalmának lefordított változata. A két generált építőelemet mindkét esetben a "+" jellel kapcsoljuk össze. Ennek jelentése egyszerűen az, hogy a tagokat a generált dokumentumban írjuk egymás után (illetve annak absztrakt alakjában a szülő elemük legyen azonos).

```
define h3 as

   trans:    elem(br) +

             elem(big,[],elem(b,[],trans(content))) +

             elem(br);
```

A H3 elem fordítási szabályát bemutató iménti összetettebb minta a fordításdefiníciós szabály középső részére vonatkozik, ez az előző számban közölt 5. ábrán látható absztrakt alakok közötti fordítás formális leírása. Egy eddig nem említett kifejezés az elem(...), amelyet ebben a szabályban használunk. Ha az elem kifejezés egy vagy két argumentumú, akkor tartalom nélküli SGML elemet kell generálnunk, azaz az absztrakt fában nincs leszármazott, illetve a karakteres alakban nincs az elemnek tartalma, ami a kezdő és záró elem közé kerülne.

Az első argumentum a generálandó elem neve, a második argumentum az elem attribútumainak a definiálására szolgál. Az attribútumokat egy speciális lista formátumban adhatjuk meg (ha nincs attribútum, akkor az [ ] üres listával jelezzük). A háromargumentumos változat segítségével tartalommal rendelkező SGML elemeket generálhatunk. Az első két argumentum jelentése azonos az előzőével, a harmadik argumentum pedig egy ugyanolyan formátumú kifejezés, amely a trans kulcsszó után állhat. Ez tulajdonképpen az elem tartalmát takarja.

Példafordítás

A fentiekben a fordításdefiníciós nyelvnek csak néhány vonását mutattuk be. Hasonló konstrukciók segítségével sokféle fordítás leírható ezen a nyelven. A következőkben egy konkrét példán keresztül is láthatjuk a módszer használatát.

Tegyük fel, hogy egy fordítóprogramot kell készítenünk, amely a már említett két nyelv, a HTML és a WML közötti fordítás elvégzésére alkalmas. Az előzőek szerint tehát a fordító megírásához csupán az egyes elemek közötti fordítási szabályokat kell definiálnunk, feltéve természetesen, hogy már rendelkezésünkre áll egy általános fordítóprogram, amely a fordításdefiníció leírását feldolgozva magát a fordítóprogramot automatikusan képes generálni.

A HTML nyelvben majdnem 100 különféle formázóelem van, így az összes fordítási szabály felírása elég nagy vállalkozás lenne, néhány szabály pedig a sok különféle eset kezelése miatt meglehetősen terjedelmes lenne, ezért csak a példadokumentumban szereplő formázóelemek fordítási szabályaival foglalkozunk. A fordítandó dokumentum egy nagyon egyszerű HTML oldal, amely egy címet, két bekezdést és egy vonalat tartalmaz. Az oldal HTML-beli leírása és megjelenítése látható a 7. ábrán.

Ennek a dokumentumnak kell megfeleltetnünk egy WML dokumentumot. Egy lehetséges megfeleltetés és annak megjelenítése látható a 8. ábrán.

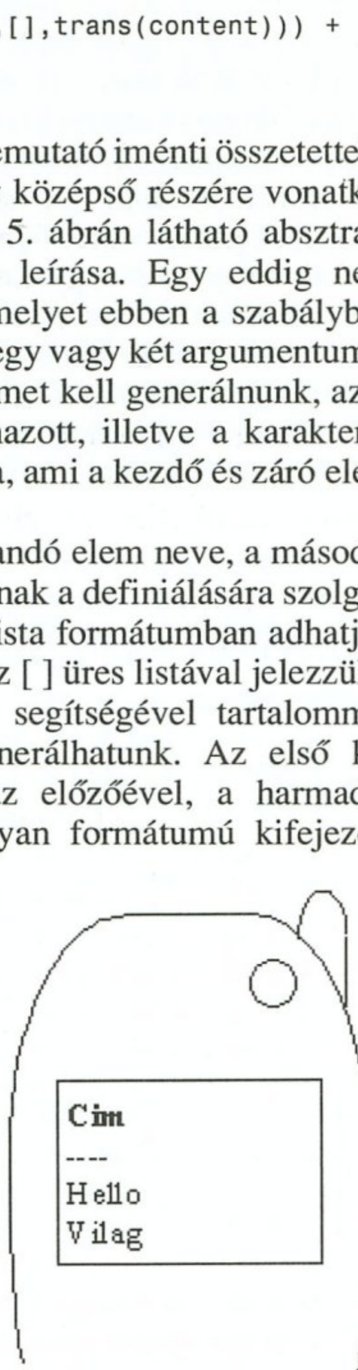
A példa alapján tehát meg kell adnunk a HTML, a BODY, a P, a H4 és a HR elemek fordításleíró szabályait. A következőkben végigkövetjük a fordítás menetét, és alkalmazásuk helyén megadjuk az egyes szabályokat. A fordítás első lépéseként elő kell állítanunk a fordítandó HTML dokumentumnak megfelelő absztrakt fát. Erre az előzőek alapján azért van szükség, hogy fordítónk gépi feldolgozásra alkalmas, szintaktikusan helyes dokumentumalakkal dolgozzon. Az előállított absztrakt fa a 9. ábrán látható együtt a fordítás eredményeként születő absztrakt fával.

A tényleges fordítás során először a HTML elem szabályát alkalmazzuk. A szabály szerint (amit most formálisan nem írunk le, hanem csak szóban adunk meg) a HTML elemnek egy WML elemet kell megfeleltetnünk, ahol a WML elem tartalma (azaz a <wml> és </wml> karaktersorozatok közötti dokumentumrész) a HTML elem tartalmának lefordított alakja. Tehát ahhoz, hogy elkészüljünk a HTML elem fordításával, le kell fordítanunk magát az elem tartalmát is. Példánkban a HTML elem összesen egy BODY elemet tartalmaz, tehát a tartalom fordítása során mindössze egy BODY elemet kell feldolgoznunk. Ennek a WML CARD elemét feleltetjük meg, aminek a tartalma a BODY elem tartalmának lefordított változata lesz.

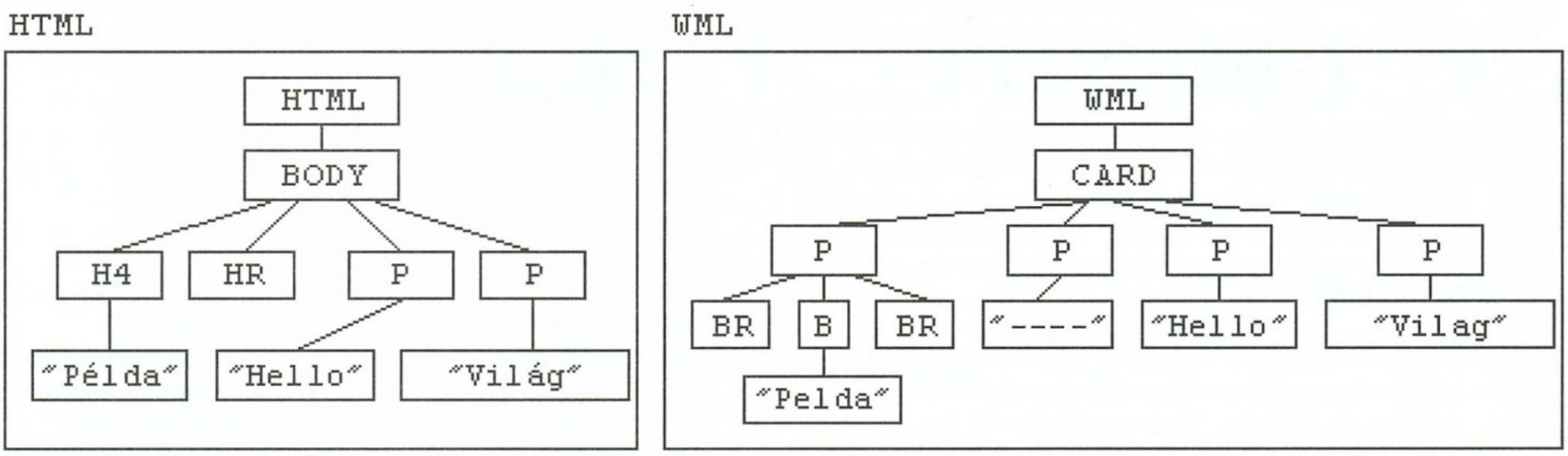
A következő fordítási lépésben már a BODY elem tartalmát dolgozzuk fel. A címkiemelésre szolgáló H4 elem fordítása során (és általában minden szöveget tartalmazó rész fordítása során) ügyelni kell arra, hogy a WML nyelv szabályai szerint szöveg a CARD elemen belül csak P (paragrafus) elemekbe zárva fordulhat elő. Tehát meg kell vizsgálnunk először, hogy CARD elembe fogjuk-e generálni a lefordított alakot, vagy nem. A két eset fordítása nagyon hasonló, abban térnek csak el egymástól, hogy ha CARD elembe generálunk, akkor még egy P elemet is generálni kell, aminek a tartalma már ugyanaz lesz, mint ellenkező esetben a lefordított alak.

Ebben a fordításban a H4 elem tartalmának lefordított alakját egy B elembe rakjuk (amely a félkövér kiemelés

```
<wml>
  <card>
  <p>
    <br/>
    <b>Pelda</b>
    <br/>
  </p>
  <p> ---- </p>
  <p> Hello </p>
  <p> Vilag </p>
  </card>
</wml>
```



8. ábra. A 7. ábrának megfelelő
WML dokumentum és megjelenítése



9. ábra. Az 7. és 8. ábra dokumentumainak megfelelő absztrakt fák

jelölésére szolgál). A kapott B elem elé és után is generálunk egy sortörést (BR elem). A H4 tartalmának fordításakor csak egyszerű karaktersorozatot kell fordítanunk (nem tartalmaz formázóelemet), aminek a fordítása tulajdonképpen önmaga, kiszűrve a WML-ben nem engedélyezett karaktereket.

Ezek után a HR elemet (vonalat) kell lefordítani. Ennek a megfelelője egy '-' karakterekből álló egyszerű vonal, kiegészítve azzal, hogy itt is vizsgálnunk kell, hogy a szülőelem CARD volt-e, és ha igen, akkor még egy P elemet is generálni kell.

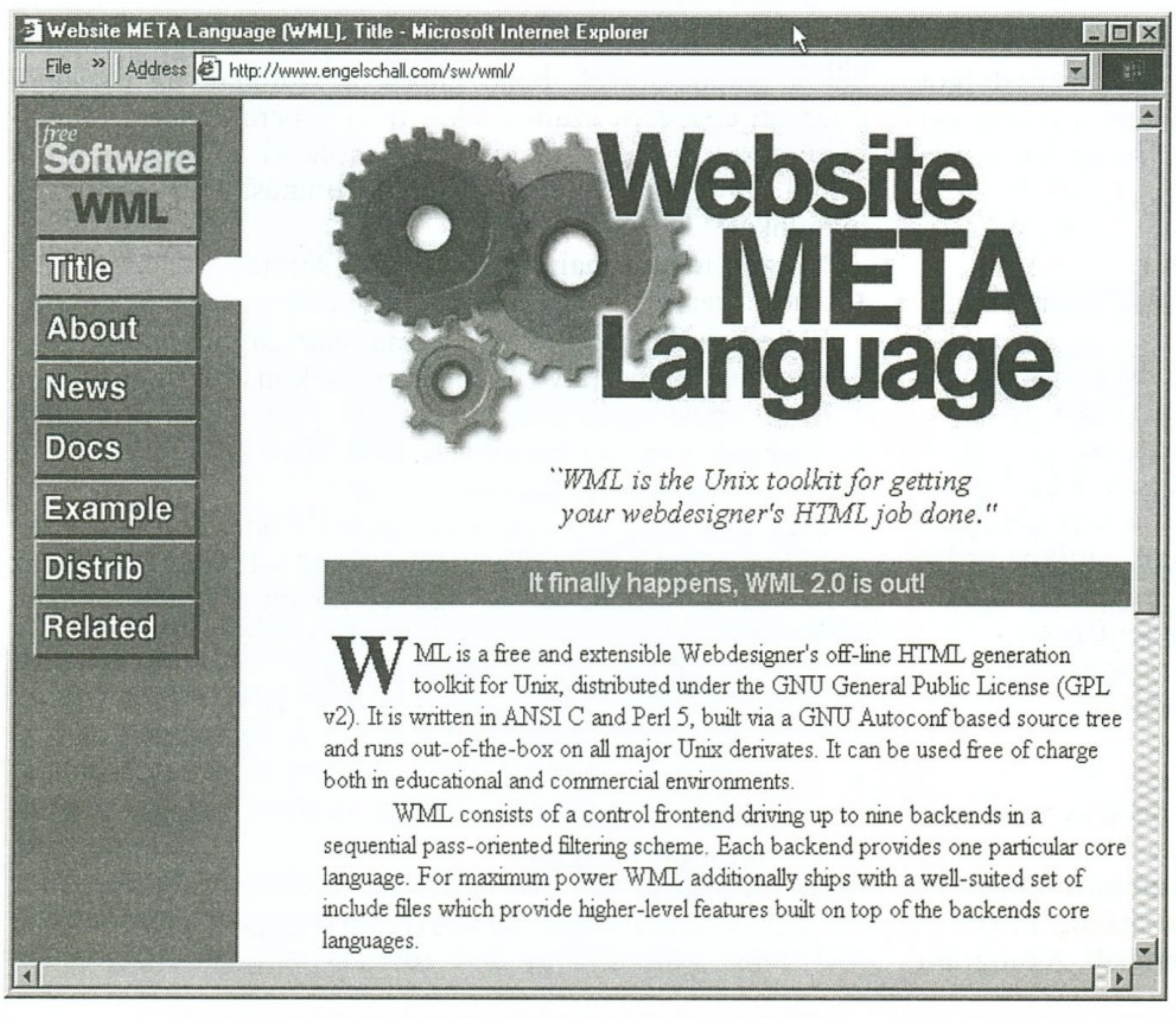
Következik a P elem fordítása, ami igen egyszerű: egy P elemet feleltetünk meg neki, ahol a WML-beli P elem tartalma a HTML-beli P tartalmának a lefordított változata lesz. Itt esetünkben ismét csak egyszerű karaktersorozatokról van szó. A következő bekezdés (paragrafus) jelzésére szolgáló P elem fordításakor is hasonlóan kell eljárni.

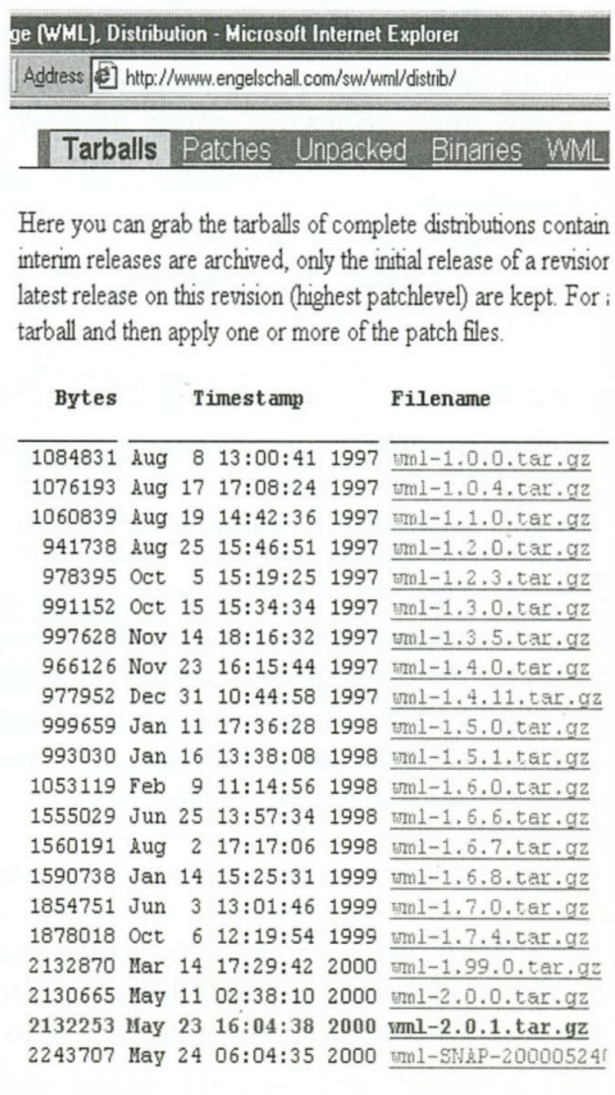
Az utolsó P elem lefordításával befejeződött a BODY elem lefordítása, hiszen megkaptuk tartalmának lefordított változatát. Ugyanígy befejeződött a HTML elem fordítása is, tehát az egész dokumentum fordításának is a végéhez értünk. Magát a fordítást a 9. ábrán látható HTML fának mélységi bejárásával végeztük el, és ennek során állítottuk elő a WML nyelvű dokumentumnak megfelelő absztrakt fát, amely szintén látható az ábrán. A fordítás utolsó lépéseként végül elő kell állítani a WML dokumentum karakteres alakját, alapul véve a generált absztrakt fa alakot. A példában szereplő WML dokumentum alakja a 8. ábrán látható.

* * *

Ez a kétrészes írás összefoglalt egy fordításmódszertant, amelynek segítségével SGML nyelvek közötti konverziót lehet formális módon definiálni. Alapja egy TDK dolgozat, amely a Budapesti Műszaki Egyetem Villamosmérnöki és Informatikai Karának TDK konferenciájára készült. A módszertan alapján olyan általános fordítóprogram készíthető, amelynek használatával könnyen lehet további SGML nyelvekhez igazodó fordítóprogramot készíteni, meglévő programok esetén pedig a feldolgozott SGML nyelv változásait egyszerűen lehet követni.

Lente Zsolt
lente@inf.bme.hu





Java tanfolyam — 6. rész

Vezérlési szerkezetek

Az if-else utasítás

A program írásakor nem tudhatjuk előre, hogy a felhasználó melyik lehetőséget választja a felajánlottak közül, és programunk futásának irányt szabnak a betáplált kezdeti feltételekből adódó különböző eredmények is. Ezért gyakran elágazásokat kell létrehoznunk, és csak futásidőben fog eldőlni, hogy a végrehajtás melyik ágon szalad végig. Ehhez használhatjuk az if-else vezérlési szerkezetet. Az if magában is állhat, ilyenkor a következő alakú:

```
if (kifejezés) utasítás
```

A kifejezésnek a kiértékelés után mindig boolean típusú eredményt kell visszaadnia. Ha az eredmény igaz (true), akkor a vezérlés az utasításra kerül, ha nem (az eredmény hamis, azaz false), akkor a program futása az if szerkezetet követő első utasításon folytatódik. Az általános forma:

```
If (feltétel) {
   //kódblokk
}
else if (feltétel) {
   //kódblokk
}
else {
   //kódblokk
}
```

Kódblokknak az utasítások azon csoportját nevezzük, amelyeket egységnek tekintünk. Ezek a blokkok egymásba is ágyazhatók. Az if, else if, else funkcióját a magyar fordítás is jól érzékelteti: ha a feltétel teljesül, tedd ezt; máskülönben: ha ez a feltétel teljesül, akkor hajtsd végre ezt az utasítást; másképp (végül): feltétel nélkül csináld ezt!

A for ciklus

Ezt a ciklust akkor használhatjuk, amikor ugyanazt a dolgot egymás után többször végre akarjuk hajtani, majd adott lépésszám után abbahagyjuk a végrehajtást. Általános alakja a következő:

```
for (kezdeti érték megadása,
      a feltétel kiértékelése, frissítés) {
   végrehajtanó kód
}
```

Kódkonvenció: A kódblokk kezdetét mutató kapcsos nyitó { zárójelet mindig ugyanannak a sornak a végére tesszük, amelyikben a for kulcsszó van. A lezáró kapcsos } zárójelet pedig azonos szintre tesszük a for kulcsszóval. (Itt a lapban a behatárolt hasábszélesség sortörései miatt a kapcsos nyitó zárójel néha átkerül a következő sor végére, amikor pedig „üres" kapcsos lezáró zárójelek követik egymást, azokat helytakarékosságból egy sorba, egymás mögé írjuk.)

Feladat: Nézzük meg az Application05 program ciklusát, majd írjunk más for ciklusokat a minta alapján!

```
for (int i = 0; i < angle; i++) {
   System.out.println("i = "+i);
   System.out.println("radians = "+Math.toRadians(i));
}
```

Az Application05 program szám argumentumot fogad el a parancssorról, majd nullától a megadott számig kiírja az egész számokat, és azok radiánná alakított értékét. A program futtatásakor használjuk a –cp opciót, ha problémánk adódik! Például: java –cp c:\ratio Application05

Megjegyzések:

1. Kezdeti érték megadása: az i változó értékét nullára állítjuk, azaz a ciklus nullával kezdődik (int i = 0).

2. A feltétel kiértékelése: a ciklus mindaddig végrehajtódik, amíg az i változó értéke kisebb, mint az angle változóban tárolt literál értéke (i < angle). A kiértékelés mindig a ciklus végrehajtása után következik be. Amikor a feltétel nem teljesül, azaz a kiértékelés hamis értéket ad vissza, akkor a ciklus véget ér.

3. Frissítés: az i változó értékét mindaddig változtatjuk, amíg a kívánt értéket el nem érjük (i++). A frissítés minden ciklus végén bekövetkezik.

A literálok konstans értékek, melyeket változókhoz rendelhetünk. A for ciklus után nemcsak kódblokk, hanem egyetlen utasítás is állhat:

Az Application05a osztály

```
class Application05a {
   public static void main(String[] args) {
      if (args.length > 0) {
         Integer wrapper = new Integer(args[0]);
         if (wrapper.intValue() == 10) {
            System.out.println("n == 10");
            for (int count = 0;
               count < wrapper.intValue(); count++)
               System.out.println("count = "+count);
         } else {
            System.out.println("n != 10");
            for (int count = 20;
               count > wrapper.intValue(); count--)
               System.out.println("count = "+count);
         }
      } else System.out.println("Kell argumentum!");
   } }
```

Feladat: Az Application05a osztály szám argumentumot kér a parancssorról, majd ennek a számnak az értékétől függően egyszerű számsorokat ír ki a parancssorra. Tanulmányozzuk a kódot, és következtessük ki belőle, hogy mit csinál a program! Majd a program lefuttatásával ellenőrizzük magunkat!

Nézzük meg a main metódus első sorát!

```
Integer wrapper = new Integer(args[0]);
```

Ebben a sorban adjuk át a parancssori argumentumot egy Integer burkolóosztály (wrapper class) konstruktorának, ami a String típust egész számmá alakítja. Futtatáskor próbaképpen adjunk meg egy bármilyen, nem szám szövegkonstanst! Milyen hibaüzenetet kapunk, és miért?

Figyeljük meg a count++ és count– – operátorokat! Ezek a count = count + 1; vagy a count = count – 1; értékadásoknak a rövidített formái. A ++ eggyel növeli a count értékét, ahányszor csak rákerül a vezérlés, míg a – – eggyel csökkenti. A fenti for ciklust így is írhattuk volna:

```
for (int i = 0; i < angle; i = i + 1) {
   ...
}
```

Két másik példa:

A ForLoop01 osztály

```
class ForLoop01 {
   public static void main(String[] args) {
      for (int i = 0; i < 10; i++)
```

```
        System.out.println("i = "+i);
    }
}
```

A ForLoop02 osztály

```
class ForLoop02 {
    public static void main(String[] args) {
        for (int i = 10; i > 0; i--)
        System.out.println("i = "+i);
} }
```

A for ciklus görbe zárójelek közötti részében nem feltétlenül kötelező a kezdeti érték és a frissítés megadása, ha a hiányukat pontosvesszővel jelöljük:

A ForLoop03 osztály

```
class ForLoop03 {
    public static void main(String[] args) {
        int i = 0;
        for (;i<0xf;) {
            System.out.println("i= "+i);
            i++;
        } } }
```

1. feladat: Keressük meg az általunk kibontott JSDK forráskódban (src.jar) az int egyszerű típus burkolóosztályát az Integer.java fájlban, majd hasonlítsuk össze azt a korábban említett Double osztállyal!

2. feladat: Másoljuk át a Double.java, Integer.java és Math.java fájlokat egy üres mappába, majd írjuk be a javadoc Double.java Integer.java Math.java utasítást a parancssoron! Ismerkedjünk a létrejövő fájlstruktúrával!

3. feladat: A javadoc program létrehoz egy java\lang mappát, amibe az általunk megadott három forráskód API dokumentációját teszi be. Böngészőprogrammal tekintsük meg a Double.html, Integer.html és Math.html fájlok tartalmát, majd hasonlítsuk össze azokat az src.jar azonos nevű, de .java kiterjesztésű forrásállományaival! A html fájlok talán áttekinthetőbbek.

4. feladat: Bár mi magunk is létrehozhatjuk az API dokumentációt a javadoc eszköz segítségével, erre nincs szükség, mert egy fájlban letölthetjük azt az internetről. (Lásd a JDK 1.2 Documentation függelékét a következő címen: http://java.sun.com/products/jdk/1.2/docs/index.html.)

Ha eddig még nem telepítettük volna az API dokumentációt, akkor most tegyük meg! Ez a dokumentáció tartalmazza azoknak az osztályoknak a leírását, amelyeket mi magunk is felhasználhatunk programjaink írásakor. A Java esetében választhatunk, hogy az API-t tanulmányozzuk, vagy az API-k alapjául szolgáló forráskódot.

A while ciklus

Nagyon hasonló a for ciklushoz. Ez is ismételten végrehajtódik mindaddig, amíg egy bizonyos feltétel fennáll. Általános alakja a következő:

```
while (feltétel) {
    végrehajtandó kód
}
```

Először mindig kiértékelődik a feltétel, ha ez igaznak bizonyul, az egész kódblokk végrehajtódik. Ezután ismét kiértékelődik a feltétel, és végrehajtódik a kód, mindaddig, amíg a feltétel nem válik hamissá. Ekkor befejeződik a ciklus.

Kódkonvenció: A kapcsos { zárójelet mindig ugyanannak a sornak a végére tesszük, amelyikben a while utasítás van, míg az utolsó } zárójelet új sorba tesszük a while kulcsszóval egy szintbe.

Az Application06 programban az i ciklusváltozót egyenlővé tesszük a parancssorról kapott számmal, amit előzőleg az angle változóba tettünk. Ezt a változót a – operátor segítségével számtanilag negáljuk (–angle). A while ciklus elején kiértékelődik a feltétel, azaz az alkalmazás megvizsgálja, hogy i nagyobb-e az angle változó értékénél. A ciklus végén pedig mindig csökkentjük a ciklusváltozó értékét eggyel (i– –).

Az Application06 osztály

```
class Application06 {
    public static void main(String[] args) {
        Double obj = new Double(args[0]);
        int angle = (int) Math.abs(obj.doubleValue());
        int i = angle;
        while (i > -angle-1) {
            System.out.println("degrees = "+i);
            System.out.println("radians =
                "+Math.toRadians(i));
            i--;
        } } }
```

Feladat: Gondolkodjunk el azon, hogy miért kell egyet levonnunk a negált angle változóból (i > – angle–1)!

A do-while ciklus

Hasonló a while ciklushoz, de a feltétel kiértékelése a ciklus végén történik. Általános alakja a következő:

```
do {
    végrehajtandó kód
} while (feltétel);
```

Látható, hogy a kódblokkban lévő utasítások legalább egyszer mindig végrehajtódnak, hiszen az első kiértékelés csak a végrehajtandó kód után következik.

Kódkonvenció: A { zárójelet mindig ugyanannak a sornak a végére tesszük, amelyikben a do utasítás van. A kódblokkot záró } zárójel mögé írjuk a while kulcsszót, míg az utolsó } zárójelet egy szintbe tesszük a do kulcsszóval.

Az Application07 osztály

```
class Application07 {
    public static void main(String[] args) {
        if (args.length > 0) {
            Double obj = new Double(args[0]);
            int angle =
                (int) Math.abs(obj.doubleValue());
            int i = 0;
            do {
                System.out.println("i = "+i);
                System.out.println("radians =
                    "+Math.toRadians(i));
            } while (i++ < angle);
    } } }
```

A ++ operátor

Az Application08 programban a ++ operátor két fajtájával is találkozunk:

1. Az i++ az operátor postfix verziója, a ++ operátor az i változó mögött van.

2. A ++i a prefix verzió, azaz a ++ operátor az i változó előtt van.

Az Application08 osztály

```
class Application08 {
    public static void main(String[] args) {
        if (args.length > 0) {
            Double obj = new Double(args[0]);
            int angle = (int) Math.abs(obj.doubleValue());
            System.out.println("postfix i++:");
            int i = 0;
            do System.out.println("i = "+i);
            while (i++ < angle);
            System.out.println("prefix ++i:");
            i = 0;
```

```
do System.out.println("i = "+i);
while (++i < angle);
} } }
```

Mint már említettük, az i++ az i = i + 1 értékadás rövid formája. Ez igaz a ++i alakra is. Akkor felmerülhet a kérdés, hogy miért van szükség a kétféle verzióra? Ha a 0 paraméterrel indítjuk el a programot (java Application08 0) , akkor azt látjuk, hogy a két ciklus futásának eredménye ugyanaz, hiszen a do-while ciklus blokkjában lévő utasítások legalább egyszer végrehajtódnak:

```
postfix i++:
0
prefix ++i:
0
```

A java Application08 1 parancs kiadásakor viszont a következő eredményt kapjuk:

```
postfix i++:
0
1
prefix ++i:
0
```

Tehát a prefix operátor ciklusa mindig eggyel kevesebbszer fut le, mint a postfix verzió ciklusa. Ennek a viselkedésnek bizonyos esetekben nagy jelentősége lehet.

1. feladat: Próbáljuk ki a fenti alkalmazást más argumentumokkal is!

2. feladat: Gondolkodjunk el azon, hogy miért kapunk különböző eredményeket? A példában ++i értéke az első kiértékeléskor már 1, ezért a ciklus véget ér, az i++ értéke az első kiértékeléskor viszont még mindig 0.

Megjegyzés: A Java kétparaméteres operátorai és az egyetlen háromparaméteres operátor mind infix operátorok, azaz az operátor mindig középen van: i < angle.

A switch vezérlési utasítás

A switch szerkezet egyfajta kényelmi szolgáltatás, hiszen az if-else utasítással is megoldhatjuk ugyanazokat a feladatokat. A forráskód olvashatóságát a switch szerkezet gyakran megjavítja, más esetekben viszont az if-else szerkezet az áttekinthetőbb. Általános formája a következő:

```
switch (feltétel) {
    case érték1 :
        kódblokk1;
        break;
    case érték2 :
        kódblokk2;
        break;
    case érték3 :
        kódblokk3;
        break;
    default :
        kódblokk4;
        break;
}
```

Megjegyzések:

1. A case kulcsszó mögötti értékeknek konstans kifejezéseknek kell lenniük, és azonos típusúaknak a feltétel típusával.

2. Ezeknek az értékeknek célszerű int vagy char típusúaknak lenniük.

3. A break kulcsszó elhagyható, de ez azzal jár, hogy a program futásakor az összes case utasításon végigmegy, még akkor is, ha már megtalálta a keresett értéket. Bizonyos esetekben viszont kívánatos, hogy ez megtörténjen.

4. A kódblokkokat nem szükséges hullámos zárójelek közé tenni.

5. A default kódblokk párhuzamba állítható az if-else vezérlési utasítás else blokkjával. Akkor kerül ide a vezérlés, amikor egyetlen feltétel sem teljesül, azaz a case értékei közül egyik sem azonos a feltétellel.

Az Appliction09 program egyszerre mutat példát a switch és az if-else szerkezetekre. Az alkalmazás 1 és 5 között visszhangozza a parancssoron beírt számokat, minden más esetben kiírja a default és else kulcsszavakat:

Az Application09 osztály

```
class Application09 {
 public static void main(String[] args) {
  if (args.length > 0) {
   int argument = Integer.parseInt(args[0]);
   switch (argument) {
    case 1: System.out.println("case 1"); break;
    case 2: System.out.println("case 2"); break;
    case 3: System.out.println("case 3"); break;
    case 4: System.out.println("case 4"); break;
    case 5: System.out.println("case 5"); break;
    default: System.out.println("default"); break;
   }
   if (argument == 1) System.out.println("if 1");
   else if (argument == 2) System.out.println("if 2");
   else if (argument == 3) System.out.println("if 3");
   else if (argument == 4) System.out.println("if 4");
   else if (argument == 5) System.out.println("if 5");
   else System.out.println("else");
  } } }
```

1. megjegyzés: Akkor kerül a vezérlés a default kulcsszóra, ha egyetlen case érték sem felel meg a feltételnek. Ugyanez a helyzet az if-else szerkezettel. Ha egyik feltétel sem teljesül, akkor végül az utolsó else végrehajtódik.

2. megjegyzés: Az if utasítás után tetszőleges számú else if utasítást következhet, de csak egyetlen else.

3. megjegyzés: Ahogy már eddig is láttuk, az if-else és else blokkok elhagyhatók.

A break kulcsszó

A break utasításnak két fajtája van:

— címkézett és

— nem címkézett break.

A nem címkézett break csak ciklusokban és a case utasítások mögött használható, ha ezeken kívül próbálkozunk vele, hibaüzenetet kapunk. Ha rákerül a vezérlés, akkor kilép a ciklus vagy a switch blokkjából.

A címkézett break csak abból a blokkból ugrik ki, amelyiket címkével megjelöltünk. Ennek a blokknak nem kell ciklusnak vagy switch elágazásnak lennie.

A LabeledBreak osztály

```
class LabeledBreak {
    void doBreak(boolean labeledBreak) {
        int i = -1;
        while (i++ < 2) {
            if (!labeledBreak)
                break;
            codeBlock1:
            if (labeledBreak) {
                System.out.println("i = "+i);
                break codeBlock1;
            }
            System.out.println("labeled break");
        }
        System.out.println("break");
    }
    public static void main(String[] args) {
        if (args.length > 0) {
        LabeledBreak app = new LabeledBreak();
```

```
        if (args[0].equals("-lb")) app.doBreak(true);
        if (args[0].equals("-b")) app.doBreak(false);
    } } }
```

Magyarázat: A LabeledBreak osztály két argumentumot fogad el a parancsorról. A –lb beírásakor a vezérlés a címkézett blokkba kerül, míg a –b argumentum átadásakor a címke nélküli break megszakítja a while ciklust, és a program futása megszakad.

Feladat: Futtassuk le a LabeledBreak programot, és a kiírt üzenetekből következtessünk arra, hogy az argumentumoktól függően hogyan változik a vezérlés.

A continue kulcsszó

A continue utasításnak a breakhez hasonlóan két fajtája van:

— címkézett és
— nem címkézett continue.

A címkézett continue

A címkézett continue utasítás esetében a vezérlés a címkével ellátott ciklusra kerül.

A LabeledContinue osztály

```
class LabeledContinue {
  public boolean isTrue(int iValue,
        int jValue, int kValue) {
     int i = 0;
     there:
     do {
        System.out.println("i = "+i);
        for (int j = 0; j < jValue+1; j++) {
           System.out.println("j = "+j);
           if (j == kValue)
              continue there;
        }
        return false;
     } while (i++ < iValue);
     return true;
  }
  public static void main(String[] args) {
     if (args.length > 0) {
        int i = Integer.parseInt(args[0]);
        int j = Integer.parseInt(args[1]);
        int k = Integer.parseInt(args[2]);
        LabeledContinue app = new LabeledContinue();
        if (app.isTrue(i,j,k))
           System.out.println("minden OK!");
        else System.out.println("hiba!");
     } } }
```

Magyarázat: A LabeledContinue osztály három argumentumot kap a parancssorról, amelyeknek egész számoknak kell lenniük. A String típusú argumentumok átalakítását az Integer burkolóosztály parseInt(String s) metódusa végzi. Majd a három argumentumot átadjuk az isTrue( ) függvénynek. Ha a függvény visszatérési értéke igaz, akkor a program kiírja az üzenetet: "minden OK!", ha hamis, akkor "hiba!".

Az isTrue( ) függvény két ciklust tartalmaz: egy do-while és egy for ciklust. Az első argumentum, ami az isTrue( ) függvény első paramétere lesz, a do-while ciklus feltételkiértékeléséhez kell. Ez mondja meg az alkalmazásnak, hogy hányszor kell végrehajtani a do-while ciklust:

```
int i = 0;
do {
   ...
} while (i++ < iValue);
```

A második argumentum az isTrue( ) függvény második paramétere lesz, és a for ciklus feltételkiértékelésének adja meg a befejezési értéket, amihez még egyet hozzáadunk, hogy a ciklus annyiszor — és ne eggyel kevesebbszer — hajtódjon végre, mint amennyi a második parancssori argumentumban megadott érték:

```
for (int j = 0; j < jValue+1; j++) {
     ...
}
```

A for ciklus a do-while ciklusba van beágyazva:

```
there: // ez a címke
do {
   ...
   for (int j = 0; j < jValue+1; j++) {
      ...
      if (j == kValue) // ha a j és a kValue egyenlő,
         continue there; // folytasd a címkénél!
   }
   return false;
} while (i++ < iValue);
return true;
```

Ha a for ciklus befejeződik, a vezérlés a return false utasításra kerül, és a függvény hamis értékkel tér vissza. Ha minden jól megy, a do-while ciklus fejeződik be, és a visszatérési érték igaz lesz.

De mikor csúszik ki a vezérlés az isTrue( ) függvényből? Figyeljük meg, hogy a do-while ciklust a there: címkével jelöltük meg, és azt, hogy a for ciklus blokkjában van egy feltételes kiértékelés. Ez megvizsgálja, hogy a j változó aktuális értéke megegyezik-e a parancssorról kapott kValue értékével. Ha igen, akkor a continue there utasítás következtében a vezérlés a there címkével megjelölt ciklusra kerül. Ez egyben a for ciklus futásának megszakadását is jelenti, mivel az nem hajtódik végre jValue+1-szer, hanem csak kValue-szor.

Ha azonban a j és a kValue nem egyenlő, akkor a vezérlés nem éri el a continue there utasítást, a for ciklus rendben befejeződik, és a return false utasítás hatására a függvény visszatér. Ez akkor következik be, ha a parancssorról kapott kValue értéke nagyobb, mint a j változóban lévő literál. A do-while ciklus ilyenkor egyszer fut le, míg a for ciklus teljes egészében befejeződik.

1. feladat: Futtassuk le a LabeledContinue alkalmazást különböző argumentumokkal! Például írjuk be a következő parancsot és argumentumokat:

```
java LabeledContinue 3 2 1
java LabeledContinue 5 4 5
```

Mikor lesz a kiírt j ciklusváltozó azonos a parancssoron megadott második számmal? Miért?

2. feladat: Hasonlítsuk össze a következő forráskódot a LabeledContinue osztály kódjával:

```
class LabeledContinue_ {
  public void doWork(int iValue,
  int jValue, int kValue) {
     int i = 0;
     do {
        System.out.println("i = "+i);
        for (int j = 0; j < jValue+1; j++) {
           System.out.println("j = "+j);
           if (j == kValue)
              System.out.println("--- "+kValue+" ---");
        }
     } while (i++ < iValue);
  }
  public static void main(String[] args) {
     if (args.length > 0) {
        LabeledContinue_ app = new LabeledContinue_();
        app.doWork(Integer.parseInt(args[0]),
```

```
        Integer.parseInt(args[1]),
        Integer.parseInt(args[2]));
    } } }
```

A nem címkézett continue

A nem címkézett continue visszaadja a vezérlést a ciklus feltételkiértékelésére, ami az UnlabeledContinue osztály példájában a következő:

```
(argumentLength-- > 0)
```

Itt ismételten megtörténik a kiértékelés, majd ennek eredményétől függően a ciklus folytatódik vagy megszakad.

Az UnlabeledContinue osztály

```
class UnlabeledContinue {
 public static void main(String[] args) {
  if (args.length > 0) {
   String word1 = new String(args[0]);
   int word1Length = word1.length();
   String word2 = new String(args[1]);
   int word2Length = word2.length();
   if (word1Length != word2Length) {
    System.out.println("A hossz nem azonos!");
    System.exit(0); // break;
   }
   int i = word1Length;
   while (i-- > 0) {
    char c1 = word1.charAt(i);
    char c2 = word2.charAt(i);
    System.out.println("i = "+i
     +" c1 = "+c1+" c2 = "+c2);
    if (c1 == c2) {
     if (i == 0)
      System.out.println("Teljesen azonosak!");
     continue;
    }
    char d1 = Character.toUpperCase(word1.charAt(i));
    char d2 = Character.toUpperCase(word2.charAt(i));
    System.out.println("i = "+i
     +" d1 = "+d1+" d2 = "+d2);
    if (d1 != d2) {
     System.out.println("Nem azonosak!");
     break;
    }
    System.out.println("Majdnem azonosak!");
 } } } }
```

Magyarázat: Az alkalmazás két szót fogad el a parancssorról, majd összehasonlítja őket. Első lépésként átadja az argumentumokat egy String típusú változónak:

```
String word1 = new String(args[0]);
String word2 = new String(args[1]);
```

A szöveg típus a Javában teljes értékű osztály, aminek metódusai vannak. Ilyen metódus a length( ), amely a szöveg változóban található 16 bites unikód karakterek számát adja vissza egy int típusú változóban, ami példánkban a word1Length és a word2Length változó:

```
int word1Length = word1.length();
int word2Length = word2.length();
```

A két szó hosszának összehasonlításával rögtön kiszűrhetjük a nem egyenlő szavakat:

```
if (word1Length != word2Length) {
   System.out.println("A hossz nem azonos!");
   System.exit(0); // break;
}
```

Ha a két szó hossza nem egyenlő, először hibaüzenetet írunk ki a standard kimenetre ("A hossz nem azonos!"), majd a System osztály exit(int status) metódusával leállítjuk a programot. A 0 státusz, azaz állapot paraméter normális leállást jelent. Ha valami hiba következtében szakítjuk meg a Java Virtuális Gép futását, akkor nullától eltérő állapot paramétert adunk meg, ami általában –1 lehet.

Az azonos hosszúság azonban még nem jelent azonos szavakat, ezért egy while ciklust hozunk létre, amely a szavak végének irányából minden betűt összehasonlít. A while ciklus feltételkiértékeléséhez az első szó hosszát használjuk, amit előzőleg átadtunk az i változónak:

```
int i = word1Length;
```

Az i változó értékét eggyel csökkentjük, valahányszor rákerül a vezérlés. Ezt mindaddig csináljuk, amíg az összes betűt össze nem hasonlítottuk.

```
while (i-- > 0) {
   // minden betű összehasonlítása a cikluson belül
}
```

A szavaknak mindig az azonos pozícióiban lévő betűit kell összehasonlítanunk. Ehhez deklaráljuk a c1 és c2 char típusú változókat, majd az = értékadó operátorral elhelyezzük bennük a word1 és a word2 változók betűit, egyiket a másik után:

```
char c1 = word1.charAt(i);
char c2 = word2.charAt(i);
```

Ehhez a String osztály charAt(int index)lenyomatú metódusát használjuk. Ez a függvény a String típusú objektumok i-edik helyen lévő karakterét adja vissza. A legelső karaktert a 0 index keresi ki, a másodikat az 1, míg a legutolsót a szöveg hossza–1 index.

A while ciklus az i változó kezdeti értékét rögtön csökkenti eggyel, így a charAt( ) függvény a legutolsó betűt adja vissza, majd eljuthatunk egészen a legelső, azaz a 0 index értékig. Itt jön a címke nélküli continue utasítás haszna. Megvizsgáljuk a c1 és c2 karaktereket, és a ciklust csak akkor folytatjuk, ha a két betű azonos. Az alábbi utasítást tehát így fordíthatjuk: ha c1 egyenlő c2-vel, akkor folytasd (a ciklus végrehajtását)!

```
if (c1 == c2)
   continue;
```

Ha c1 nem egyenlő c2-vel, akkor itt ismételten befejezhetnénk a program futását, hiszen a két szó nem bizonyult egyenlőnek. De ha a kis- és nagybetűket nem különböztetjük meg, akkor megnézhetjük, hogy a nagybetűssé alakított karakterek azonosak lesznek-e. Ehhez a Character burkolóosztály public static char toUpperCase(char ch) lenyomatú metódusát használjuk, ami a paraméterben átadott char egyszerű típust nagybetűvé alakítva adja vissza:

```
char d1 = Character.toUpperCase(word1.charAt(i));
```

Most ellenőrizzük, hogy a nagybetűssé átalakított karakterek azonosak-e:

```
if (d1 != d2) {
   System.out.println("Nem azonosak!");
   break;
}
```

Ha nem egyenlőek, akkor kiírjuk a "Nem azonosak!" üzenetet a standard kimenetre, és a break utasítással kilépünk a while ciklusból.

Megjegyzés: A Character.toLowerCase( ) statikus metódussal lehet a betűket kisbetűssé alakítani a fentihez hasonló módon.

1. feladat: Futtassuk le az UnlabeledContinue alkalmazást a következő argumentumokkal:

java UnlabeledContinue Ratiosoft Ratiosoft (A két szó teljesen azonos)
java UnlabeledContinue Ratiosoft ratiosoft (A két szó azonos, de az egyik csupa kisbetűvel van írva)
java UnlabeledContinue Ratiosoft Ratioxoft (A két szó nem azonos)
java UnlabeledContinue Ratiosoft ratio (A két szó hossza eltérő)

Tanulmányozzuk, hogy milyen üzeneteket ír ki az alkalmazás a fenti esetekben, és gondolkodjunk el azon, hogy miért ezekre került a vezérlés!

2. feladat: Cseréljük ki a System.exit(0) utasítást a break utasítással, majd fordítsuk le újra a programot! Miért kapunk hibaüzenetet a fordítás során? Miért fogadja el a javac fordító a másik break utasítást a "Nem azonosak!" üzenetet kiíró kódblokkban?

A return kulcsszó

A return szintén vezérlési utasításnak tekinthető, hiszen használatával befolyásolhatjuk a program futását, az általunk meghatározott ponton léphetünk ki az adott kódblokkból.

A return utasítás szintén kétféle lehet:
— függvényekben értéket visszaadó return,
— eljárásokban üres értéket visszaadó return.

Függvény

Ha egy függvény definiálásakor nem adunk meg visszatérési értéket, holott azt a lenyomatban deklaráltuk, akkor hibaüzenetet kapunk:

```
public boolean function() { }
```

Példánkban Return required at end of boolean function( ) (azaz Return kell a boolean függvény végén) hibaüzenettel állt le a javac fordító. A függvény minimális alakja a következő lehet:

```
public boolean function() {
   return true;
}
```

Eljárás

Mivel egy eljárásban nincs visszatérési érték, vagy pontosabban a visszatérési érték void, a return kulcsszó mögé nem írunk semmit:

```
public void showExplicitReturn() {
   for (int i = 1; i < 3; i++)
   System.out.println("i = "+i);
   return; //explicit return
   }
```

Bár a fordító elfogadja az explicit return deklarációkat az eljárások végén, azokra valójában nincs szükség:

```
public void showImplicitReturn() {
   for (int i = 1; i < 3; i++)
   System.out.println("i = "+i);
}
```

Az explicit return kulcsszónak olyankor van jelentősége, amikor elágazást akarunk létrehozni, azaz bizonyos feltételek teljesülésekor meg akarjuk szakítani az eljárás futását:

```
public void procedure(boolean doNothing) {
   if (doNothing) return;
   System.out.println("Zavarhatsz! Most dolgozom.");
   for (int i = 0x0F; i > 0x0C; i--)
   System.out.println("i = "+i);
}
```

A fenti procedure( ) nevű eljárásban a doNothing (azaz ne csinálj semmit!) paraméter igaz értéke esetén semmi sem történik, a vezérlés a return utasításra kerül.

Megjegyzés: Láthatjuk, hogy a function és a procedure a Javában nem fenntartott szavak.

A ReturnDemo osztály

```
class ReturnDemo {
   public String function(boolean doNothing) {
      String message;
      if (doNothing) return message = "Pihenj!";
      else return message = "Dolgozz!";
   }
   public void procedure(boolean doNothing) {
      if (doNothing) return;
      System.out.println("Zavarhatsz! Most dolgozom.");
      for (int i = 0x0F; i > 0x0C; i--)
      System.out.println("i = "+i);
   }
   public void showImplicitReturn() {
      System.out.println("Ne zavarj! Alszom!");
   }
   public void showExplicitReturn() {
      System.out.println("Ne zavarj! Ez fusi!");
      for (int i = -3; i < 5; i++)
      System.out.println("i = "+i);
      return;
   }
   public static void main(String[] args) {
      if (args.length > 0) {
         System.out.println("Dolgozzak?");
         ReturnDemo app = new ReturnDemo();
         boolean doNothing = false;
         if (args[0].equals("-p")) doNothing = true;
         System.out.println(app.function(doNothing));
         app.procedure(doNothing);
         if (doNothing) app.showImplicitReturn();
         if (!doNothing) app.showExplicitReturn();
} } }
```

Feladat: A ReturnDemo program a –p argumentumot kéri be a parancssorról. Minden más beírása esetén a doNothing változó értéke hamis lesz, mivel ezt az alapértéket állítottuk be (doNothing = false;). Figyeljük meg, hogy a return használata hogyan befolyásolja a program futását, majd írjunk a programhoz új eljárásokat és függvényeket a minták alapján!

Kivételkezelő utasítások

A try-catch-finally utasítások szintén a vezérlési szerkezetekhez tartoznak. Bővebben lásd később!

A végtelen ciklus

Az InfiniteLoop program látszólag hibátlan. A javac eszköz lefordítja, és végrehajtja azt a feladatot, amit elvárunk tőle. Bizonyos bemeneti értékek esetén mégis végtelen ciklusba kerül, és csak a DOS-ablak bezárásával lehet leállítani. Ez a program kicsi és könnyen áttekinthető, de ízelítőt ad abból, hogy milyen hibák léphetnek fel a nagy és bonyolult felépítésű programok esetében. A fejlesztő teszteli azt a programrészt, amelyet neki kell megírnia, s az akár tökéletesen is működhet az adott körülmények között, de összerakva a nagy programot, váratlan hibák léphetnek fel. Ilyen végzetes hiba lehet a nullával való osztás is.

```
class InfiniteLoop {
   public static void main(String[] args) {
      if (args.length > 0) {
         for (int i = 0; i < 15; i++) {
            System.out.println("i = "+i);
            if (i == Integer.parseInt(args[0]))
               i = Integer.parseInt(args[1])-1;
         } } } }
```

Magyarázat: Az InfiniteLoop alkalmazás két egész számot fogad el a parancssorról. Majd nullától számol az első parancsori argumentumig, itt megszakad a ciklus, és a második parancssori argumentumtól folytatódik egészen 14-ig. Írjuk be például:

```
java InfiniteLoop 2 12
```

Feladat: Keressük meg, hogy milyen parancssori argumentumok beírásakor kerül végtelen ciklusba a fenti program! Mi ennek az oka?

Szaló István
ratiosoft@freemail.c3.hu

E SZÁMUNK HIRDETŐI

Mikrobazár

A Mikrobazár rovatban a nem kereskedelmi célú egyéni hirdetések közlése ingyenes. A kereskedelmi célú apróhirdetések tarifája gépelt soronként (azaz 60 karakterenként) 300 forint. A terjedelem alapján így kiszámított összeget kérjük átutalni az Új Alaplap Kiadói Kft számlájára (OTP, 11706016–20788599), vagy feladni postai utalványon a kiadó címére (1539 Budapest, Pf. 571), és feltüntetni, hogy „Új Alaplap, apróhirdetés". A befizetést igazoló szelvény másolatát — a hirdetési szöveggel együtt — a szerkesztőséghez (a kiadóéval azonos címre) küldjék el.

Szerzői jogokat sértő szoftverhirdetéseket nem közlünk le.

Bármilyen típusú szöveg fordítását vállalom angolról magyarra, magyarról angol nyelvre, illetve vállalom kiadványok látványtervezését, szerkesztését is. Cím: Lachner Zoltán, 1195 Budapest XIX., Jáhn Ferenc u. 14/a. Telefon: 357-0308.

OBJECTS 2.0 — objektumorientált programozás CLIPPER-ben: www.tar.hu /proxima.

Adatmentés CD-re, streamerre; winchesterről, floppyról. Ugyanitt beszerzési tanácsadást, hálózattervezést és programkészítést is vállalok. Cím: Kovács Lajos, 1031 Budapest III., Vízimolnár u. 10. IV/33.

Alaplapcsere, memória-, winchester- és floppybővítés a helyszínen. MegaSoft. Telefon: 295-5085.

Stúdiómban megbízhatóan, ellenőrzötten lefordítom angol, német, francia és magyar nyelvről/nyelvre műszaki és közgazdasági folyóiratok cikkeit, hardver- és szoftverleírásait. Áfás számlát állítok ki. Cím: Szász György, 1035 Budapest III., Kórház u. 25. Tel.: 368-4874.

Súlyosan mozgáskorlátozott, bal kéz hiánnyal született gyermekem részére (akinek felnőtt korában megélhetési forrása a számítástechnikai ismeret lehet) **keresek olcsó** számítógépkiegészítő hardver- (RAM, bővítőkártya, CPU stb.) és szoftverelemeket. Ajándékozás esetén adójóváírás megoldható. Cím: Szőke János, 5440 Kunszentmárton, Tanya 604. Telefon: (56)707-123.

Seagate Medalist 4321 (ST34321A)-hoz keresek hibátlan elektronikát, illetve Seagate Medalist ST 31276A-hoz hibátlan mechanikát olcsón. Telefon: 06-30-216-1605.

Eladó az Új Alaplap 1999. október – 2000. májusi száma CD nélkül, 260 Ft/db áron. Postázás esetén plusz postaköltség. Pesten átadás megoldható: Telefon: 284-7228. e-mail: szjozsi@mailbox.hu

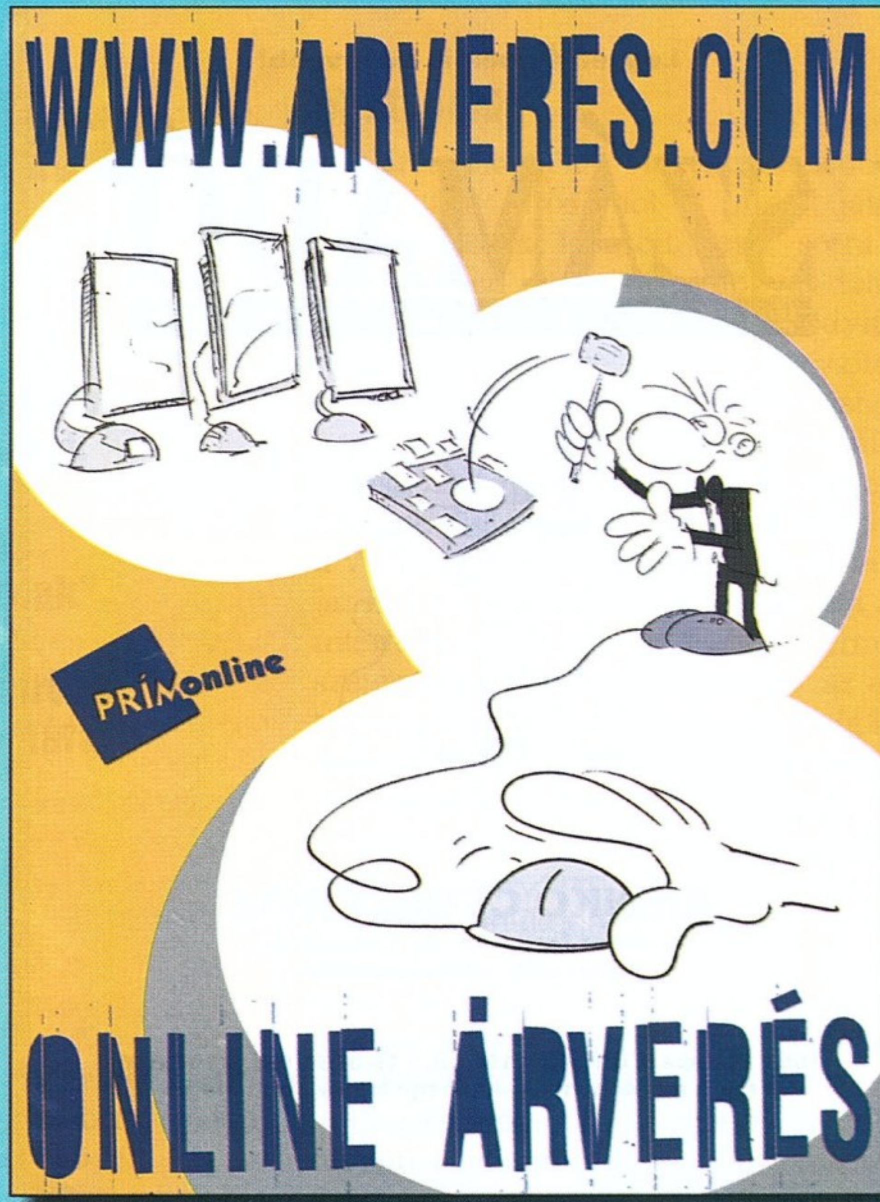
Fényképek szkennelése, archiválása CD-n. SCI-FO Egyéni Cég. Tel.: 320-4702.

PC-karbantartó, felhasználó asszisztens 6 év gyakorlattal, vállalkozói igazolvánnyal **állást keres.** Részletes önéletrajz kérhető: Telefon: Cseppentő Árpád, 312-8172.

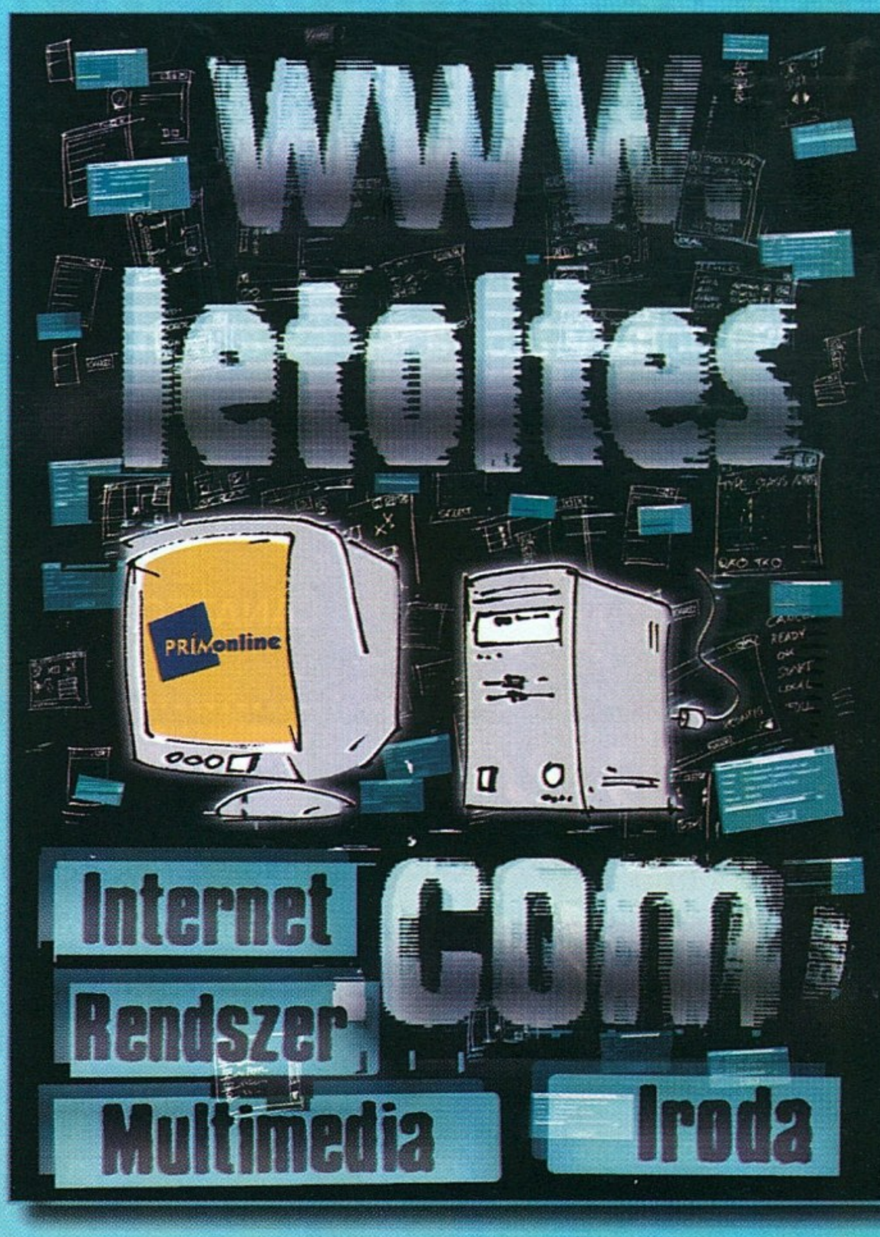
JÚLIUSBAN A HÓNAP TÉMÁJA:

ELEKTROKRÁCIA









INFORMÁCIÓKÉRÉS: 14 ▼

Gazdasági informatika

Új alapozó sorozat

A ComputerBooks mostanáig főleg a Műegyetem tájékáról szerezte be szerzőgárdájának utánpótlását, sőt témáinak tekintélyes részét is. A Műegyetem élen jár a számítástechnika oktatásában, és sok fejlesztésből is kiveszi a részét. Az új szoftverek állandóan „vizsgáznak" a műegyetemi oktatásban, az oktatógárda rengeteg tapasztalatot szerez azok használhatóságáról, előnyeiről és gyengeségeiről. Mindez nemcsak a hallgatók számára jelent értéket, hanem a nagy nyilvánosság számára készülő köteteken keresztül a szélesebb olvasóközönségnek is. A jelekből ítélve most kiszélesedik a kiadó szerzői gárdája.

A megújult Közgáz (hivatalos nevén: Budapesti Közgazdaságtudományi és Államigazgatási Egyetem, BKÁE) intenzívebben bekapcsolódott a számítástechnikai tankönyvek készítésébe. Speciálisan a gazdasági informatika az a terület, amelyet a BKÁE saját profiljának tekinthet. Ezt hivatott szolgálni az új sorozat.

Mire jók a hibák?

Az első két kötet tükrözi azt a koncepciót, amelyet a közgazdasági egyetem számítástudományi tanszékén megfogalmaztak:

(1) Tömören, lényegre törően foglalni össze a legfontosabb információkat.

(2) A bemutatott szoftvereszközöknek a gyakorlatban való használatára koncentrálni.

(3) A műszaki részletekbe csak annyira belemenni, amennyire az a gyakorlati munkához nélkülözhetetlen.

(4) Sokkal nagyobb teret szentelni a problémák feltárásának, mint a hasonló témájú könyvekben az szokásos.

Balogh Gábornak igen jól sikerült megvalósítani ezeket a célokat a Visual Basicről és az Excel programozásáról írott könyvében. Ez a sorozat többi kötetének elkészítéséhez is példamutató lehet. Tömör és precíz a fogalmazása, kevés szóval jól megragadja a lényeget. A szerző attól sem riad vissza, hogy egy-egy alkalommal „behúzza a csőbe" az olvasót, hogy a későbbiekben frappáns módon elvezesse a hiba megtalálásához.

A programok első változatai rendszerint sok hibát tartalmaznak. Az elkövetett hibákból azonban rengeteget lehet tanulni: ugyanazt másodszorra sokkal kisebb valószínűséggel követi el az ember. A programozás mesterségének elsajátításához is a hibák legyűrésén keresztül vezet az út. Ezért fontos annak a képességnek a kifejlesztése, hogy viszonylag hamar felismerjük a hibák okát.

A szerző mindenekelőtt az elbizakodottság veszélyére hívja fel a figyelmet. Senki ne képzelje, hogy a Visual Basic roppant egyszerűnek tűnő fejlesztői környezetében nem lehet hibákat elkövetni. Ezért fontos, hogy visszafelé is tudjunk következtetni, és megtaláljuk a hibák forrását.

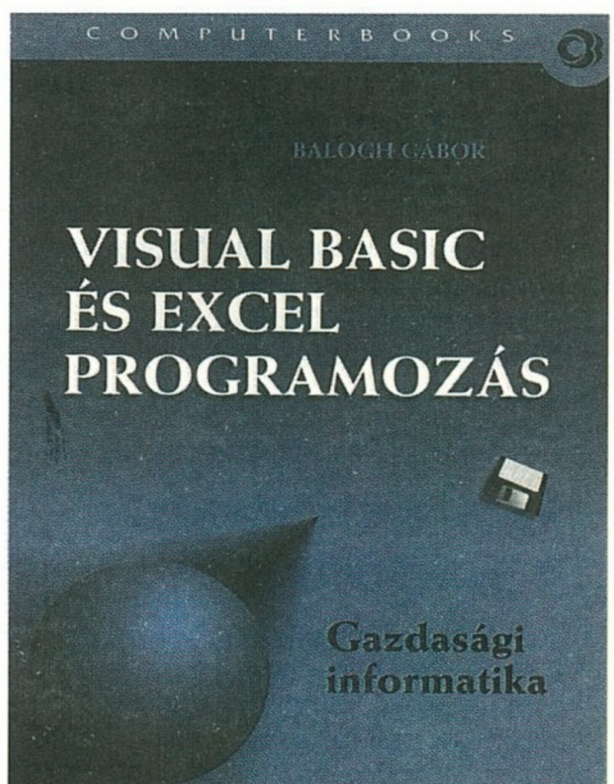
A Visual Basic valóban ideális környezet lehet azoknak, akik viszonylag egyszerű rendszert akarnak fejleszteni, és nem nagyon zavarja őket, ha programjuk nem működik villámgyorsan. Csak a nagy számításigényű feladatok végzésekor válik feltűnővé a lassúság, amikor újra meg újra interpretálni kell a p nyelvű utasításokat. Két nagyságrendnyi lassulás is bekövetkezhet ilyenkor. Igaz, a Visual Basic 6.0 már képes natív kódot is előállítani, de az csak akkor futásképes, ha rendelkezésünkre áll az MSVBVM60.DLL modul, amely biztosítja a Windows erőforrás-menedzsereivel a kommunikációt. A p kód használatának is megvannak az előnyei: elsősorban az, hogy könnyen megvalósíthatóvá teszi a hordozhatóságot. Gyakorlatilag a p kód speciális gépfüggetlen standardnak tekinthető. A neve is erre utal, p = a portabilitás (hordozhatóság, átvihetőség) rövidítése.

Mintapélda

Mintapéldaként legrészletesebben egy „itt a piros, hol a piros" játék elkészítésének lépéseit veszi sorra a szerző, ezen mutatja be, hogyan lehet Visual Basicben megtervezni és kivitelezni egy korrekt módon elkészített programot. Megismerhetjük a leírásból, hogy a fejlesztőkörnyezet aktív támogatásával hogyan készíthető el a program specifikációja és a programterv, mi a VB projekt létrehozásának csínja-bínja. Ezután már érdemes hozzáfogni az interfész form elkészítéséhez, természetesen ha birtokában vagyunk az interfész elemek kezelésére szolgáló technikának. Majd elkezdődhet magának a programkódnak a megírása, az eseménykezelő eljárások felhasználásával. Ekkor már sikerélményhez is juthatunk: működik a programunk. Ha valamit mégis figyelmen kívül hagytunk, az a későbbiek során derül ki, amikor átfogó tesztelésnek vetjük alá programunkat.

Amikor már látjuk, hogy milyen technika alkalmazására van szükség feladataink megoldásához, következnek a formmal kapcsolatos legfontosabb tudnivalók, és megismerhetjük a beépített vezérlések használatát, a nyomógomboktól kezdve a szöveges vezérlődobozokig és az egeres vezérlésig. (A lemezmellékleten mindegyikhez találhatók példák is.) Kiderül a „control" kifejezés kétértelműsége az angol nyelvű számítástechnikai zsargonban, és a szerző a „control array" megfelelőjeként javasolja a magyar terminológiában a kevésbé félreérthető „objektumtömb" elnevezést.

A Visual Basic programnyelv rövid ismertetése után a kötetben viszonylag



részletes eszmefuttatás található a hibakeresésről és hibakezelésről, a felhasználható eszközök és módszerek ismertetésével. Ezt követi néhány fontos algoritmus bemutatása, a menükészítés technikája, és néhány szó az adatbázisokról is.

Az Excel

Talán meglepő, hogy a címben ígért másik témát, az Excel programozását a szerző elég könnyedén intézi el, de ebbe a rövid fejezetbe is belefér az Excel programozási környezetének, valamint az Excel objektumainak ismertetése. Aki az előzményekből jól megértette az OOP lényegét és felhasználási technikáját, az ebből a kevésből is ért. Megtanulhatjuk a párbeszédes vezérlőelemek programozását, szóba kerül a UserFormok, az eseménykezelési eljárások, a beépített függvények és a felhasználói függvények készítése és használata. A lemezmelléklet programozási példákkal szolgál a Solver-szerűen komplex funkciók Excel-beli megvalósításához. (Mellesleg lapunk 9 részes sorozatot közölt az Excel programozásáról. Álló Géza: Excel felsőfokon, Új Alaplap, 1998. június — 1999. február.)

Tőzsdejáték

A könyv lemezmellékletére került rá a szerző egyik tanítványának, Tóth Péternek egyik szellemes és értékes alkotása, a Tőzsdejáték. Ezzel szinte élőben lehet „tőzsdézni", anélkül, hogy bárki kockáztatná saját vagyonát. A körülmények valóban életszerűek: választhatunk a brókercégek közül, rábízhatjuk pénzünket és részvényeinket, megbízást adhatunk neki részvények vételére és eladására. Mindezt annak ismeretében, hogy a cégek hogyan szerepelnek az egyes napokon, az egyes részvényeknek mekkora a forgalma, milyen árakon történik az üzletkötés, mik a nyitó- és záróárak. Egy-egy kiválasztott részvényhez grafikont is készít a program.

Brókerünknek egészen kifinomult módon adhatunk megbízást. A limitáras változatnál köthetjük ki, hogy maximum mennyiért vagyunk hajlandók megvenni a részvényt, eladáskor pedig, hogy mennyit kérünk érte minimálisan. Megszabható például, hogy csak akkor adják el, ha a részvény piaci árfolyama a megadott szint alá csökken. Vételnél hasonló megszorítások lehetnek. Bejelölhetjük továbbá, hogy megbízásunk részteljesítéssel vagy anélkül érvényes.

Fontos tulajdonsága a programnak, hogy a játék sebességét változtatni lehet. A valóságban ugyanis a tőzsdézők idejük nagy részében várakoznak. Vagy arra, hogy emelkedjenek, vagy arra, hogy essenek az árak, s hogy egyáltalán történjen valami.

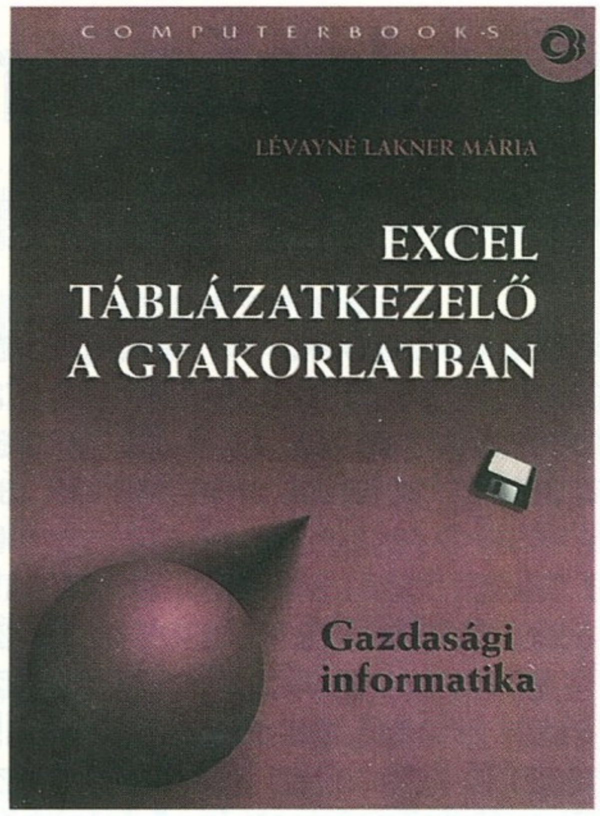
A várakozás azonban hosszadalmas, és gyakran unalmas is. Ha úgy látjuk, hogy semmi érdekes nem történik, felgyorsíthatjuk a történéseket, máskor pedig lelassíthatjuk. A 21 fokozatú skálán a leggyorsabb fokozatban gyakorlatilag semmi beavatkozási lehetőségünk nincs, szinte egyetlen pillanat alatt zajlik le az egész napi kereskedés. A leglassúbbnál viszont elérhető a tőzsde illúziója: a napi majdnem 3 órás kereskedés a gépen ugyanannyi idő alatt zajlik le.

Ha brókerünkkel nem vagyunk megelégedve, másikat kereshetünk. Nincs viszont beleépítve a programba az a szituáció, amikor a brókercég csődbe megy. Elég nehéz is lenne ezt a helyzetet életszerűen szimulálni. Mi lehetne például a kritériuma annak, hogy rosszul dolgozott? És milyen információkkal kellene erről rendelkeznie a programnak? A véletlenszerű csőd ugyan könnyen beépíthető lenne, de minek.

Aki pedig ihletet ebből a programból is merítve tovább tudja fejleszteni a tőzsdejáték modelljét, keresse fel a szerzőt. Lehet, hogy a kötet következő kiadásában már az ő programját is mellékelni fogják…

Van másik?

Az előző, imponáló módon elkészített kötet mellett a második kissé halványabbnak tűnik, főleg a rutinosabbak számára. A kezdők véleménye talán pozitívabb, mert a könyvben benne van mindaz, amit a kezdőknek az Excel



használatáról illik tudni. Korrekt módon leírja a szerző az alapfunkciókat és az alapfogalmakat, példákkal sem fukarkodva. Más leírásoktól eltérően nagyobb teret enged a statisztikai feladatok megoldásának és a pénzügyi elemzéseknek, de talán még így sem eleget. Matematikai alkalmazásként a mátrixműveleteket tárgyalja, a lineáris egyenletrendszerekkel és optimalizálási feladatokkal együtt. Bőségesebb az adatbázisként definiált adattáblák bemutatása. Az Excel programozásával kapcsolatban a szerző a makrók használatára fordít nagyobb figyelmet.

Hasznosak a függelékben a megoldásra szánt feladatok. A könyvben szereplő mintapéldák induló adatai és megoldásai a lemezen megtalálhatók. Könnyű válogatni a témakörönként csoportosított feladatok közül, ezáltal a lemez példaanyaga vizsgára való készüléshez vagy vizsgáztatáshoz is jól felhasználható.

Sok hely maradt azonban kihasználatlanul a floppyn. Fel lehetett volna tenni függelékként például a túlzott magyarításnak áldozatul esett Excel függvényeinek angol és magyar elnevezését, szintaktikai leírását, kiegészítve felhasználási példákkal.

A szerzőt bizonyára szorongatta a szűkre szabott terjedelem. Talán jobb lenne, ha a sorozat további kötetei a haladóbbak igényeit ki tudnák elégíteni. Gondoljunk arra, hogy az egyetemekre ma már sokkal több számítástechnikai ismerettel kerülnek be a diákok, és sokan tisztában vannak az Excel köznapi használatával. Ők az egyetemtől már a közgazdasági alkalmazás specifikumait várják. Vagy nekik másik tankönyv készül?

Vargha Dénes

Balogh Gábor:

Visual Basic és Excel programozás

„Gazdasági informatika" sorozat
ComputerBooks, 2000
176+4 oldal, 1900 Ft
(floppymelléklettel)

Lévayné Lakner Mária:

Excel táblázatkezelő a gyakorlatban

„Gazdasági informatika" sorozat
ComputerBooks, 2000
148+4 oldal, 1200 Ft
(floppymelléklettel)

ELŐFIZETÉS

Az 2000/......... számtól kezdődően előfizetem

az Új Alaplap című CD-mellékletes havi számítástechnikai folyóiratot

...... példányban ❑ 1 évre, ❑ 1/2 évre.
Az éves előfizetési díj: 6996,– Ft (Ez az összeg az áfát is tartalmazza.)

❑ **Számlát** kérek (banki átutalással fizetek). ❑ **Befizetési csekket** kérek.

Név: ....

(Cég:) ....

Cím: ....

Irányítószám, helység: ....

Dátum:

/aláírás/

APRÓHIRDETÉS

Kérem, hogy az Új Alaplap következő számának Mikrobazár rovatában az alábbi szövegű apróhirdetést jelentessék meg. (A túloldalon ismertetett feltételeket tudomásul veszem.)

....

....

....

....

....

....

....

....

(Maximális terjedelem 300 betű.)

INFORMÁCIÓKÉRÉS

Az Új Alaplap mostani számában megjelent hirdetések közül az általam itt megjelölt kódszámúakhoz részletesebb információt kérek a hirdető cégektől.

01	02	03	04	05	06	07	08	09	10	11	12
13	14	15	16	17	18	19	20	21	22	23	24
25	26	27	28	29	30	31	32	33	34	35	36
37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70	71	72
73	74	75	76	77	78	79	80	81	82	83	84
85	86	87	88	89	90	91	92	93	94	95	96

Új Alaplap, 2000. júniusi szám. Beküldési határidő: 2000. június 30.

Belföldön díjmentesen is feladható

ÚJ ALAPLAP

VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest

Feladáskor kérjük bérmentesíteni!

FELADÓ:

Név:...

Cím:...

Helység: ....

Irányítószám:..

Telefon: ....

❑ A hirdetés egyéni és egyedi jellegű, ezért kérem ingyenes megjelentetését. Kijelentem, hogy annak tartalma nem sérti senki szerzői jogát.

❑ A hirdetés kereskedelmi célt szolgál. Mellékelem a soronként (60 karakterenként) 300 forintnak megfelelő összeg átutalásáról az igazoló szelvény másolatát. A címzett: Új Alaplap, 1539 Budapest, Pf. 571, illetve átutalásnál az OTP 11706016-20788599 számlaszámra.

/aláírás/

Bélyeg helye

ÚJ ALAPLAP

VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest

FELADÓ:

Feladáskor kérjük bérmentesíteni!

a) EGYÉNI

Név: ....

Cím: ....

Helység, ir.sz.: ....

b) CÉGES

Név: ....

Cég: ....

Cím: ....

Helység, ir.sz.: ....

Telefon: ....

/aláírás/

Bélyeg helye

ÚJ ALAPLAP

VI., Dózsa György út 84/b
Postafiók 571
1539 Budapest



INFORMÁCIÓKÉRÉS: 20 ▲